VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018 Bc. Denis Herinek

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

CERTIFIKAČNÍ AUTORITA
CERTIFICATION AUTHORITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Denis Herinek**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Václav Zeman, Ph.D.**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Denis Herinek ***ID:*** 164279
Ročník: 2 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Certifikační autorita

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte a popište služby, které zajišťuje certifikační autorita, autorita časových razítek a s nimi související infrastruktura veřejných klíčů (PKI). Navrhněte a realizujte systém, který umožní uvedené služby zajišťovat a demonstrovat tak funkčnost PKI. Při realizaci preferujte využití OpenSource SW.

DOPORUČENÁ LITERATURA:

[1] DOSTÁLEK, Libor. - VOHNOUTOVÁ, Marta. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. Vyd. 2. Brno : Computer Press, 2010. 544 s. ISBN 978-80-251-2619-6.

[2] Zákon č. 297/2016 Sb. Zákonu o službách vytvářejících důvěru pro elektronické transakce. Sbírka zákonů České republiky. 2016, částka 115, s. 4466-4504. ISSN 1211-1244.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 21.5.2018

Vedoucí práce: doc. Ing. Václav Zeman, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

V súčastnosti existuje mnoho dostupných služieb na internete, ktoré vyžadujú väčšiu bezpečnosť a dôveru. Infraštruktúra verejných kľúčov je využívaná najmä v oblastiach, kde je požadovaná vyššia úroveň autentifikácie, dôvernosti alebo integrity prenášaných správ. Je ťažké si predstaviť fungovanie elektronického bankovníctva alebo elektronického podpisu dôležitých dokumentov bez PKI. Na internete je dostupných množstvo open-source realizácií PKI vytvorených užívatelmi. Digitálne certifikáty sú vydávané certifikačnou autoritou. Diplomová práca pozostáva z open-source realizácie certifikačnej autority a autority časových pečiatok na demonštráciu služieb, ktoré poskytuje.

KĽÚČOVÉ SLOVÁ

Certifikačná autorita, infraštruktúra verejných kľúčov, PKI, digitálny certifikát, digitálny podpis, eIDAS, časová pečiatka

ABSTRACT

There is a lot of available services on the internet those need to be more secured and trusted. Public key infrastructure is used in sectors where are higher expectations in case of authentication, integrity and confidentality. It is almost impossible to imagine how internet banking or electronic signatures of important documents would work without PKI. There is a lot of open-source realisations of PKI created by users. Digital certificates as a part of PKI are issued by certificate authorities. This diploma thesis consists of open-source realisation of certificate authority and timestamping authority to demonstrate services which they provide.

KEYWORDS

Certificate authority, public key infrastructure, PKI, digital certificate, digital signature, eIDAS, timestamp

HERINEK, Denis. *Certifikační autorita*. Brno, 2017, 60 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Václav Zeman, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju diplomovoú prácu na tému „Certifikační autorita“ vypracoval(a) samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce doc. Ing. Václav Zeman, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k zlepšeniu práce.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ÚVOD

Zaisteniu bezpečnosti je vo všetkých odvetviach kladený veľký dôraz. Na internete je dostupných množstvo služieb, ktorých zabezpečenie a dôveru treba zvyšovať. Infraštruktúra verejných kľúčov je využívaná najmä v oblastiach, kde je požadovaná vyššia úroveň autentifikácie, dôvernosti alebo integrity prenášaných správ. Jej praktické využite je možné nájsť v elektronickom podpisovaní a pečatení úradných dokumentov, zabezpečenom pripojení (napr. internetové bankovníctvo) alebo v rôznych aplikáciách a niektoré z týchto služieb sú v súčasnosti nazývané ako služby vytvárajúce dôveru. Existuje množstvo programátorov, ktorí vytvárajú open-source softvéry, pomocou ktorých si užívatelia môžu vytvoriť svoju vlastnú infraštruktúru verejných kľúčov (PKI) a to napr. v súkromnej alebo vo firemnej sieti. Technické riešenie PKI môže byť realizované viacerými spôsobmi. Je možné využitie decentralizovaného modelu, v ktorom sa nenachádza žiadna všeobecne uznávaná certifikačná autorita. Diplomová práca sa však bude venovať PKI na základe štandardu X.509, založeného na vlastníctve digitálneho certifikátu, vydávaného certifikačnou autoritou. Certifikačné autority musia mať dôveru u užívateľov a spĺňať určité požiadavky.

Cieľom diplomovej práce bude preštudovanie a popísanie služieb, ktoré musia byť zaistené pre správne fungovanie certifikačnej autority, autority časových pečiatok a infraštruktúry verejných kľúčov. K certifikačnej autorite neodmysliteľne patrí aj registračná autorita, ktorá môže pracovať samostatne alebo byť jej súčasťou a tvorí dôležitý prvok pri vytváraní certifikátov. Ďalším cieľom bude navrhnutie a realizovanie systému, ktorý umožní vytváranie, spravovanie a používanie digitálnych certifikátov a vytváranie časových pečiatok s použitím vybraného alebo vybraných open-source softvérov.

V úvode práce je uvedený krátky náhľad do legislatívnych zmien, ktoré sa v českej legislatíve prejavili následkom nariadenia Európskej únie. Obsahuje niektoré zákony a vyhlášky, ktoré regulujú a definujú činnosti v oblasti infraštruktúry verejných kľúčov a na zákony, ktoré sa venujú službám zaisťujúcich dôveru.

Druhá kapitola obsahuje úvod do infraštruktúry verejných kľúčov a sú v nej uvedené informácie o elektronickom podpise, elektronickej pečati, zabezpečenej komunikácii, digitálnych certifikátoch a popis služieb, ktoré zaisťuje certifikačná autorita a autorita časových pečiatok.

Tretia kapitola bola venovaná porovnaniu dostupných možností na vytvorenie PKI vyskúšaním a výberom najvhodnejšieho open-source softvéru.

Štvrtá kapitola obsahuje detaily inštalácie certifikačnej autority a softvéru pre realizáciu autority časových pečiatok. V piatej kapitole bola nakonfigurovaná a vyskúšaná funkčnosť služieb zaisťovaných certifikačnou autoritou ako napr. vytvorenie certifikátu, vytvorenie zoznamu zrušených certifikátov (CRL) alebo skontrolovanie

on-line stavu certifikátu pomocou OCSP protokolu.

V šiestej kapitole je uvedený návod na vytvorenie autority časových pečiatok pomocou softvéru SignServer a vyskúšanie funkčnosti vrátane vytvorenia a overenia elektronického podpisu.

Posledná kapitola obsahuje vytvorenie laboratórnej úlohy, ktorá môže slúžiť študentom na oboznámenie sa s infraštruktúrou verejných kľúčov, s fungovaním certifikačnej autority a autority časových pečiatok.

1 SLUŽBY VYTVÁRAJÚCE DÔVERU

Pri výmene informácií na medzinárodnej úrovni postupne vznikalo množstvo dohadov spôsobených odlišnosťami v zákonoch a predpisoch jednotlivých krajín, ktoré definovali služby vytvárajúce dôveru, preto bolo potrebné tieto odlišnosti zredukovať. Aj v českej legislatíve sa digitálne certifikáty rozlišujú na komerčné a kvalifikované. Vydávanie kvalifikovaných certifikátov závisí od zákonov v danej krajine a je kontrolované príslušnými úradmi. Od 1. júla 2016 nadobudlo účinnosť „Nařízení Evropského parlamentu a Rady (EU) č. 910/2014 ze dne 23. července 2014 o elektronické identifikaci a službách vytvářejících důvěru pro elektronické transakce na vnitřním trhu a o zrušení směrnice 1999/93/ES“. Toto nariadenie je známejšie pod názvom eIDAS (electronic IDentification, Authentication and trust Services).

1.1 eIDAS

Ako napovedá názov, od ktorého je odvodená skratka eIDAS, nariadenie sa týka oblasti elektronickej identifikácie, autentifikácie a služieb vytvárajúcich dôveru. Keďže sa jedná o nariadenie a nie len o smernicu, tak platnosť v Českej republike nastala nadobudnutím účinnosti nariadenia a to 1. júla 2016. Novým zavedeným prvkom zo strany Európskej únie sú napr. elektronické pečate. Úpravy sa týkajú aj kvalifikovaných certifikátov pre elektronické podpisy, avšak vlastníci týchto kvalifikovaných certifikátov vydaných pred nadobudnutím účinnosti nariadenia EÚ ich podľa článku 51 môžu využívať do skončenia platnosti. Väčšia časť nariadenia sa venuje aj kvalifikovaným prostriedkom pre vytváranie elektronických podpisov (napr. USB tokeny, čipové karty) [1].

Čipovou kartou rozumieme plastickú kartu, ktorá obsahuje čip (bezkontaktné obsahujú aj anténu) a pre jej použitie je zväčša nutné použiť čítačku čipových kariet. USB token sa pripája k počítaču priamo USB portom. Zväčša je možné tieto hardvérové zariadenia zakúpiť u príslušnej certifikačnej autority [2]. V minulosti bolo dôležité súkromný kľúč na týchto hardvérových zariadeniach uchovávať mimo dosah ostatných osôb, no nová vyhláška eIDAS vytvára možnosti, kedy na základe žiadosti užívateľa môže vytvoriť elektronický podpis pomocou súkromného kľúča užívateľa niekto iný. [1].

1.2 eIDAS a štátna legislatíva

V Českej republike boli prijaté 2 tzv. adaptačné zákony a súvisiace zmenové zákony týkajúce sa služieb vytvárajúcich dôveru. Prvým je adaptačný zákon č. 297/2016 Sb.

„Zákon o službách vytvářejících důvěru pro elektronické transakce“, s ktorým súvisí zmenový zákon č. 298/2016 Sb. „Zákon, kterým se mění některé zákony v souvislosti s přijetím zákona o službách vytvářejících důvěru pro elektronické transakce, zákon č. 106/1999 Sb., o svobodném přístupu k informacím, ve znění pozdějších předpisů, a zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů“.

Druhým adaptačným zákonom je zákon č. 250/2017 Sb. „Zákon o elektronické identifikaci“. S ním bol prijatý aj súvisiaci zmenový zákon č. 251/2017 Sb. „Zákon, kterým se mění některé zákony v souvislosti s přijetím zákona o elektronické identifikaci“.

1.2.1 Zákon č. 297/2016 Sb.

Tento adaptačný zákon je účinný od 19. septembra 2016 a v náväznosti na predpis Európskej únie upravuje niektoré postupy poskytovateľov služieb vytvárajúcich dôveru, niektoré požiadavky na služby vytvárajúce dôveru, pôsobnosť Ministerstva vnútra v oblasti služieb vytvárajúcich dôveru a sankcie za porušenie povinností v oblasti služieb vytvárajúcich dôveru [3].

1.2.2 Zákon č. 250/2017 Sb.

Tento zákon je platný od 19. júla 2017, ale účinnosť bola stanovená až na 1. júla 2018. [4]. Podľa ministerstva vnútra Českej republiky priamo naväzuje na eIDAS a umožní jednoduchý, bezpečný a štátom zaručený spôsob preukazovania totožnosti užívateľa internetu a pomôže k rozvoju a jednoduchému využívaniu online služieb verejnej správy [5].

1.3 Zmeny týkajúce sa certifikačných autorít

K zvýšeniu dôvery vo vnútroštátny trh a na podporu používania služieb vytvárajúcich dôveru, EÚ odporučila zaviesť pojmy ako „kvalifikované služby vytvárajúce dôveru“ a „kvalifikovaný poskytovateľ služieb vytvárajúcich dôveru“, ktoré budú sprevádzané stanovením požiadavkov a povinností zaistenia vysokej úrovne bezpečnosti. Preto bolo potrebné posúdiť či daná autorita spĺňa nové požiadavky. Čas na splnenie požiadaviek mali príslušné autority do 1. júla 2017, inak by sa už po tomto termíne nepovažovali za kvalifikovaného poskytovateľa služieb vytvárajúceho dôveru (článkom 51 sa ním do 1. júla 2017 považovali) [1].

1.3.1 Zoznam poskytovateľov v Českej republike

Ministerstvo vnútra Českej republiky zverejnilo 1. septembra 2017 na svojej internetovej stránke aktualizovaný zoznam kvalifikovaných poskytovateľov služieb vytvárajúcich dôveru a poskytovaných kvalifikovaných služieb vytvárajúcich dôveru. Medzi kvalifikovaných poskytovateľov služieb vytvárajúcich dôveru patrí: První certifikační autorita, a. s., Česká pošta, s. p., eIdentity a. s. a Software602 a. s. [6]. Zoznam kvalifikovaných služieb vytvárajúcich dôveru, ktoré poskytujú obsahuje tab. 1.1.

Tab. 1.1: Prehľad poskytovaných kvalifikovaných služieb vytvárajúcich dôveru.

Poskytovateľ	Poskytované služby
První certifikační autorita, a. s.	Vydávanie kvalifikovaných certifikátov pre elektronické podpisy; Kvalifikovaná služba overovania platnosti kvalifikovaných elektronických podpisov a pečatí; Vydávanie kvalifikovaných certifikátov pre elektronické pečate; Vydávanie kvalifikovaných elektronických časových pečiatok;
Česká pošta, s. p.	Vydávanie kvalifikovaných certifikátov pre elektronické podpisy; Vydávanie kvalifikovaných certifikátov pre elektronické pečate; Vydávanie kvalifikovaných certifikátov pre autentifikáciu internetových stránok; Vydávanie kvalifikovaných elektronických časových pečiatok;
eIdentity a. s.	Vydávanie kvalifikovaných certifikátov pre elektronické podpisy;
Software602 a.s.	Kvalifikovaná služba overovania platnosti kvalifikovaných elektronických podpisov a pečatí; Kvalifikovaná služba uchovania kvalifikovaných elektronických podpisov a pečatí;

2 INFRAŠTRUKTÚRA VEREJNÝCH KĽÚČOV

PKI (public key infrastructure) je systém opatrení, ktoré sú spojené s vydávaním, distribúciou, správou a používaním digitálnych certifikátov. Tieto certifikáty obsahujú verejný kľúč, ktorý je verejne dostupný. Súkromný kľúč musí zostať chránený pred osobami, ktoré by jeho znalosť mohli zneužiť. Využívaním PKI je zaistených niekoľko základných bezpečnostných služieb alebo funkcií. Autentifikácia zabezpečuje, že druhá komunikujúca strana je tým, za koho sa vydáva. Zaistenie integrity znamená, že dáta neboli úmyselne alebo neúmyselne zmenené. Dôvernosť zaisťuje, že vymieňané dáta nemohli byť prečítané treťou stranou. Nepopierateľnosť zaisťuje, že podpísanie alebo šifrovanie určitej správy nemôže autor po odoslaní poprieť. Pomocou PKI je možné zaistiť okrem základných bezpečnostných služieb aj ďalšie služby založené na týchto základoch ako napr. zabezpečenie komunikácie, kedy je po autentifikácii možné dohodnutie šifrovacieho algoritmu a ustanovenie kľúčov [7].

2.1 Kryptografické mechanizmy využívané v PKI

V PKI sú využívané viaceré kryptografické mechanizmy. Patria medzi ne aj jednocestné funkcie a metódy asymetrickej kryptografie, pre ktoré je typické používanie dvojice kľúčov na rozdiel od symetrickej kryptografie, kde je používaný súkromný kľúč, ktorý poznajú obe komunikujúce strany. V asymetrickej kryptografii je verejný kľúč verejne známy a súkromný kľúč si musí vlastník adekvátne chrániť.

2.1.1 Hašovacia funkcia

Haš alebo inak nazývaný aj odtlačok, je výstupný reťazec jednocestnej hašovacej funkcie. Táto matematická funkcia vytvorí z ľubovolne dlhého textu krátky reťazec konštantnej dĺžky. Jednocestnou funkciou sú nazývané algoritmy, ktoré nie sú výpočtovo náročné, ale je výpočtovo náročné k výstupnému reťazcu nájsť pôvodný text. Pri drobnej zmene pôvodného textu sa očakáva výrazne iný výsledok [2]. V kryptografii sú používané tzv. kryptografické hašovacie funkcie. U niektorých starších algoritmov sa postupne darí nájsť pôvodné texty s rovnakým odtlačkom, preto sa postupne nahradzuje ich používanie a kedysi najpoužívanejší algoritmus SHA-1 nahrádzajú kvôli bezpečnosti algoritmy pod súhrnným označením SHA-2. Tieto algoritmy boli navrhnuté Národnou Bezpečnostnou Agentúrou v USA (NSA) a vydané Národným inštitútom pre štandardy v USA (NIST) [8]. Aby bola kryptografická hašovacia funkcia použiteľná musí spĺňať 3 základné požiadavky:

- Odolnosť voči kolízii – malo by byť výpočtovo náročné nájdenie dvoch rôznych vstupných dát, ktoré majú rovnaký výstupný reťazec.

- Odolnosť voči získaniu predlohy – na základe znalosti vytvoreného odtlačku určitých dát, by malo byť výpočtovo náročné nájdenie dát, ktorých výstupom by bol rovnaký odtlačok.
- Odolnosť voči získaniu inej predlohy – na základe znalosti vstupných dát, by malo byť výpočtovo náročné nájdenie ďalších vstupných dát, ktoré by vytárali rovnaký odtlačok [9].

2.1.2 Digitálny podpis

Digitálny podpis je vhodný pre zaistenie integrity dát a nepopierateľnosti podpisu. Vďaka zaisteniu integrity je možné zistenie, či boli dáta od podpísania zmenené. Nepopierateľnosťou je zaistená identifikácia podpisujúcej osoby. Vďaka týmto dvom aspektom je digitálny podpis technicky ekvivalentný s ručným podpisom a za dodržania určitých podmienok môže byť legislatívne uznávaný ako kvalifikovaný elektronický podpis [10]. Digitálny podpis je vytváraný v dvoch krokoch:

- Výpočet odtlačku (hašu) z dokumentu.
- Výsledný odtlačok sa podpisuje (šifruje) súkromným kľúčom podpisujúcej osoby. Tento podpísaný odtlačok dokumentu sa nazýva digitálny podpis. Podpis nie je možné podvrhnúť, ak vlastník bezpečne manipuluje so svojím súkromným kľúčom.

Po odoslaní dokumentu spolu s digitálnym podpisom môže príjemca vykonať verifikáciu, ktorá prebieha v troch krokoch:

- Príjemca vypočíta odtlačok zo samostatného dokumentu.
- Príjemca „dešifruje“[1] prijatý digitálny podpis pomocou verejného kľúča podpisujúcej osoby.
- Príjemca porovná odtlačok, ktorý je výsledkom prvého kroku a odtlačok, ktorý je výsledkom druhého kroku. Ak sú tieto odtlačky rovnaké je to dôkaz, že podpis mohol vytvoriť len vlastník súkromného kľúča (odosielateľ) a obsah dokumentu nebol počas prenosu zmenený [2].

2.1.3 Časová pečiatka

Časová pečiatka je podobná dátová štruktúra ako digitálny certifikát, avšak zväzuje dokument s určitým časovým údajom. Časová pečiatka je digitálne podpísaná autoritou časových pečiatok (TSA). Slúži ako dôkaz, že dokument (odtlačok dokumentu), existoval v konkrétnom čase. Primárne obsahuje čas, odtlačok dokumentu, meno vydavateľa pečiatky a sériové číslo. V tejto dátovej štruktúre je udávaný svetový čas – UTC. Časová platnosť pečiatky je odvodená od platnosti certifikátu TSA.

[1] spojenie dešifruje pri digitálnom podpise nie je až tak presné.

Časová pečiatka dokumentu však neslúži len ako dôkaz o existujúcom dokumente v danom čase, ale chráni ho aj proti zmenám, pretože každá zmena dokumentu by časovú pečiatku zneplatnila [2].

2.2 Elektronický podpis a elektronická pečať

Elektronickým podpisom sa rozumejú dáta v elektronickej podobe, ktoré sú pripojené k iným dátam alebo sú s nimi logicky spojené. Elektronický podpis rozlišujeme podľa úrovne, kde najvyššiu dôveryhodnosť má kvalifikovaný elektronický podpis. Ďalším druhom elektronického podpisu je zaručený elektronický podpis, ktorý musí byť jednoznačne spojený s podpisujúcou osobou, musí umožňovať identifikáciu podpisujúcej osoby a je pripojený k podpísaným dátam spôsobom, že je možné zistiť akúkoľvek zmenu dát. Kvalifikovaný elektronický podpis je rovnocenný vlastnoručnému podpisu a ak je založený na kvalifikovanom certifikáte v jednom členskom štáte EÚ, tak sa uznáva ako kvalifikovaný elektronický podpis aj v ostatných členských štátoch a takýto typ podpisu dokazuje, že dokument bol naozaj podpísaný danou osobou - nepopierateľnosť. Pre vytvorenie takéhoto podpisu je nutné vlastniť súkromný kľúč a digitálny certifikát, ktorý slúži druhej strane na overenie pravosti elektronického podpisu. Elektronická pečať vzniká podobným spôsobom ako elektronický podpis a takisto má stanovené jednotlivé úrovne až po kvalifikovanú elektronickú pečať. Pečať by mala slúžiť ako dôkaz, že dokument vydala právnická osoba a mala by poskytovať istotu o pôvode a integrite dokumentu. Pri podpisovaní alebo pečatení dôležitých dokumentov, by malo byť zabezpečené dlhodobé uchovávanie informácií pre zabezpečenie právnej platnosti a možnej validácie bez ohľadu na budúce technologické zmeny [1].

2.3 Digitálny certifikát

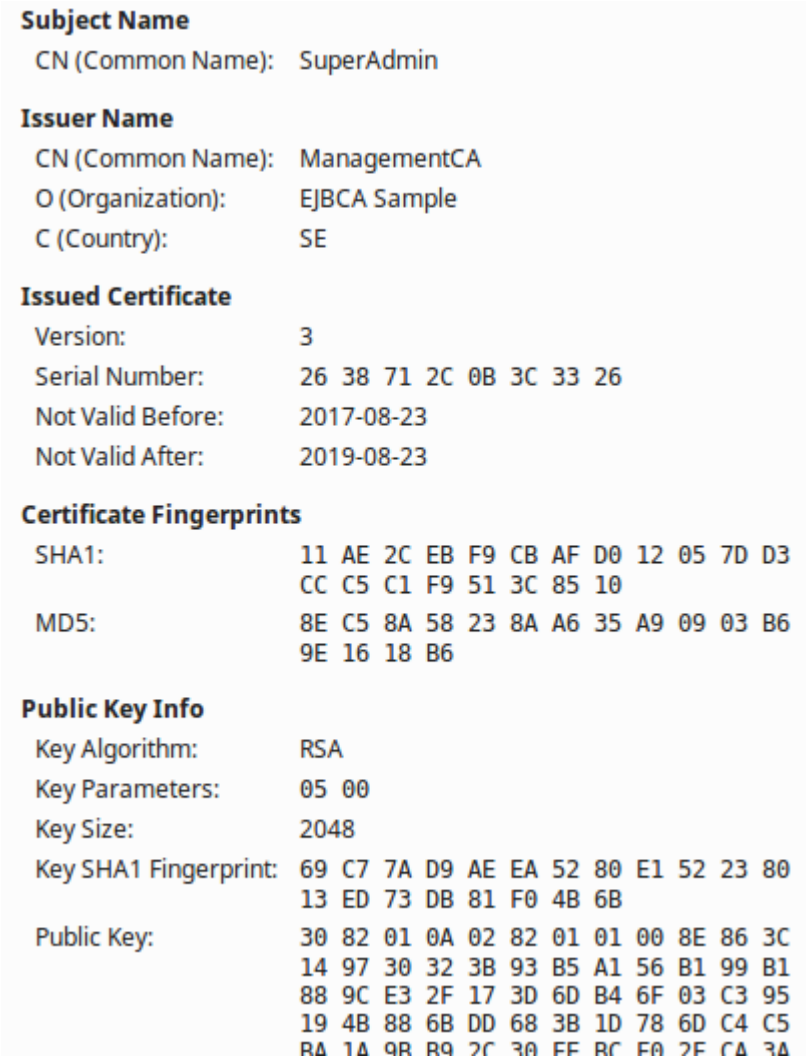
Používanie verejného kľúča si z pohľadu užívateľov vyžaduje istotu, že príslušný súkromný kľúč naozaj vlastní osoba alebo systém, ktorý ho využíva na digitálne podpisovanie alebo dešifrovanie. Potvrdzovanie vlastníkov dvojice kľúčov je riešené pomocou digitálnych certifikátov [12]. Je to elektronicky podpísaná dátová štruktúra obsahujúca verejný kľúč a identifikačné údaje držiteľa certifikátu, ktorá je často prirovnávaná k občianskemu preukazu alebo pasu [2]. Formát certifikátu je určený štandardom ITU-T X.509. V súčastnosti sa používa verzia 3. ISO/IEC, ITU-T a ANSI X9 vytvorili navyše rozšírenie tohto štandardu, ktoré umožňuje doplniť dodatočné identifikačné údaje o subjekte alebo informácie o certifikačnej politike. Keďže sa certifikáty používajú v rôznych aplikáciách a prostrediach, hlavným cieľom štandardu

bolo určiť základnú všeobecnú štruktúru certifikátu [12]. Hlavné položky certifikátu sú nasledovné:

- **Verzia certifikátu (Version)** – informuje o tom, od ktorej verzie normy X.509 bol certifikát odvodený (verzia 1,2 alebo 3).
- **Sériové číslo (Serial number)** – musí byť kladné celé číslo a pre každý certifikát v rámci certifikačnej autority aj unikátne (nemôžu byť vydané 2 certifikáty s rovnakým sériovým číslom). Z toho dôvodu položky sériové číslo a vydavateľ jednoznačne určujú certifikát.
- **Algoritmus podpisu (Signature algorithm)** – položka, ktorá špecifikuje použité algoritmy zo strany CA, pre vytvorenie elektronického podpisu certifikátu. Vždy bývajú uvedené 2 algoritmy. Jeden pre výpočet odtlačku (hašu) a druhý algoritmus je asymetrický (napr. RSA).
- **Platnosť (Validity)** – určuje platnosť certifikátu. Je rozdelená na 2 položky. „Not valid before“ určuje odkedy je certifikát platný a položka „Not valid after“ určuje, kedy jeho platnosť vyprší. Určenie doby platnosti je dôležité najmä z hľadiska bezpečnosti. Doba platnosti certifikátu musí byť omnoho kratšia ako doba potrebná k prelomeniu verejného kľúča. Certifikáty certifikačných autorít by mali mať dobu platnosti dlhšiu než je doba platnosti užívateľských certifikátov, aby nebolo nutné veľmi časté obnovovanie užívateľských certifikátov.
- **Vydavateľ (Issuer)** – meno CA, ktorá certifikát podpísala a vydala. Je potrebné, aby táto položka jednoznačne určovala CA, preto musí mať jedinečné meno (DN) v rámci všetkých CA [2]. Jedinečné meno je tvorené čiastočnými informáciami ako všeobecne známe meno (CN), krajina (C), organizácia (O), lokalita (L) atď. [12].
- **Predmet certifikátu (Subject)** – identifikuje entitu, ktorej je priradený verejný kľúč. Pri certifikátoch podľa normy X.509 verzie 3, musí byť predmet jedinečný v rámci danej CA, ktorá certifikát vydáva (certifikáty 2 osôb nemôžu mať rovnakú položku predmet). Preto môže CA vydávať rovnakej osobe certifikáty s rovnakým predmetom. Predmet môže zostať aj prázdny (prázdna sekvencia jedinečných mien), no bolo by nutné vyplniť položku „alternatívne meno“.
- **Informácie verejného kľúča (Public key info)** – túto položku tvorí samotný verejný kľúč a identifikátor algoritmu, ku ktorému je určený vrátane jeho dĺžky v bitoch.
- **Rozšírenia certifikátu (Extensions)** – patria sem informácie, ktoré neboli uvedené v hlavných položkách a je možné ich použiť len pri verzii 3. Keďže niektorým položkám aplikácie nerozumejú, boli pridané položky „závažnosť rozšírenia“, kde podľa hodnoty TRUE alebo FALSE je možné rozpoznať či sa jedná o závažné rozšírenie alebo nie. Softvér, ktorý pracuje s certifikátom musí

rozumieť všetkým závažným rozšíreniam v opačnom prípade musí certifikát odmietnuť. V rozšíreniach je definované napr. použitie kľúča, ktoré obmedzuje spôsob použitia verejného kľúča napr. na elektronické podpisovanie, zašifrovanie kľúča, zašifrovanie dát alebo podpisovanie CRL. Možné je takisto definovanie rozšíreného použitia kľúča, alternatívneho mena predmetu alebo certifikačnej politiky, ktorá obsahuje identifikátor tohto verejného dokumentu [2]. Ukážku štruktúry certifikátu je možné vidieť na obr. 2.1

Špecifickým typom certifikátu je tzv. certifikát podpísaný sám sebou a tvorca tohto certifikátu ním vytvára koreňový kľúč vlastnej certifikačnej autority [11].



Obr. 2.1: Ukážka štruktúry certifikátu (OS Ubuntu)

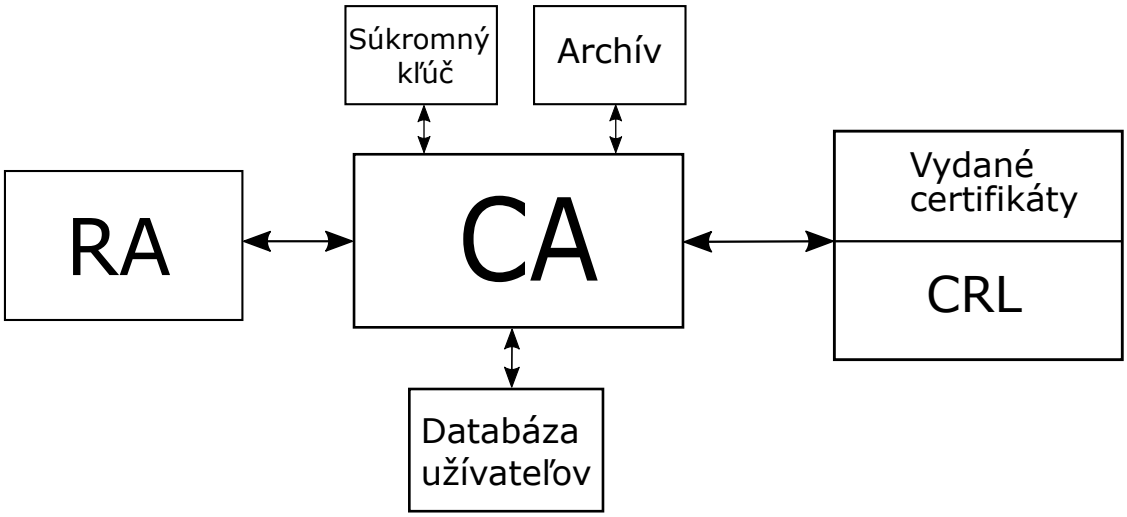
2.4 Služby zaisťované certifikačnou autoritou

Certifikačná autorita je určitý typ autority, ktorej dôveruje jeden alebo viacero užívateľov v oblasti vytvárania a prideľovania certifikátov (v ďalšom texte bude certifikačná autorita označovaná aj ako CA). Tieto certifikáty podpisuje CA súkromným kľúčom, ktorého strata by mala pre ňu fatálne následky. Môže takisto pre užívateľov vytvárať dvojicu kľúčov alebo dvojica kľúčov môže byť generovaná na strane užívateľa, ktorý ich pri žiadosti o certifikát dodá danej CA. Každá CA má verejne dostupný dokument s názvom certifikačná politika, ktorý obsahuje súbor pravidiel, ktorými sa určitá CA riadi. Odkaz na tento dokument býva zväčša uvedený na vydanom certifikáte a pomáha pri rozhodovaní či vydanému certifikátu je možné dôverovať [11]. Štruktúra CA závisí od služieb, ktoré poskytuje. Príklad štruktúry je možné vidieť na obr. 2.2. CA zaisťuje nasledujúce služby:

- Vydávanie certifikátov – CA môže vydávať rôzne druhy certifikátov. Môžu to byť komerčné certifikáty, ktoré sú najrozšírenejšie a nezávisia od legislatívy danej krajiny alebo kvalifikované certifikáty, ktoré sa viažu na legislatívu v danej krajine.
- Uchovávanie informácií – archív certifikátov po vypršaní platnosti, verejne dostupný zoznam vydaných certifikátov.
- Zrušenie certifikátu – inak povedané zneplatnenie alebo odvolanie certifikátu. Zväčša na žiadosť užívateľa napr. po kompromitácii súkromného kľúča. Tieto certifikáty sú pravidelne vydávané na zozname zrušených certifikátov.
- Online zisťovanie stavu certifikátu – overovanie platnosti certifikátu v reálnom čase.
- Obnovovanie platnosti certifikátu – obnova certifikátu, ktorého platnosť sa blíži ku koncu.

2.4.1 Vydávanie certifikátov

Jadrom CA je aplikácia, ktorá vydáva certifikáty. Certifikáty sú elektronicky podpisované súkromným kľúčom CA, ktorý je tak jej najväčším aktívom a je potrebné ho chrániť. Miesto uchovania súkromného kľúča rieši každá CA zvlášť, zväčša je to zariadenie bez prístupu na internet čím sa zvyšuje úroveň bezpečnosti. Väčšina CA využíva na sprostredkovanie vydávania certifikátov a overovanie totožnosti žiadateľov tzv. registračné autority (RA). RA môžu fungovať ako bankové prepážky, kedy je nutná osobná prítomnosť alebo ako servery, kedy komunikácia s užívateľom prebieha elektronicky. CA nevydáva certifikáty len koncovým užívateľom, ale môže vydať certifikáty aj podriadeným CA a tie môžu vydať certifikáty ďalším im podriadeným CA. Týmto sa vytvára tzv. stromová štruktúra certifikačných autorít,



Obr. 2.2: Príklad štruktúry certifikačnej autority

kde vrchol stromu tvorí koreňová CA. Ako môže vyzerať táto stromová štruktúra je zobrazené na obr. 2.3. Aby však tento proces vydávania certifikátov podriadeným CA nepokračoval, je nutné do rozšírenia certifikátu pridať obmedzenie na vytváranie podobných certifikátov. Môže sa stať, že certifikáty dvoch užívateľov boli vydané rôznymi CA, ktoré nie sú súčasťou rovnakého stromu CA, avšak tieto CA si môžu vzájomne podpísať svoje certifikáty. Tento úkon sa nazýva krížová certifikácia.

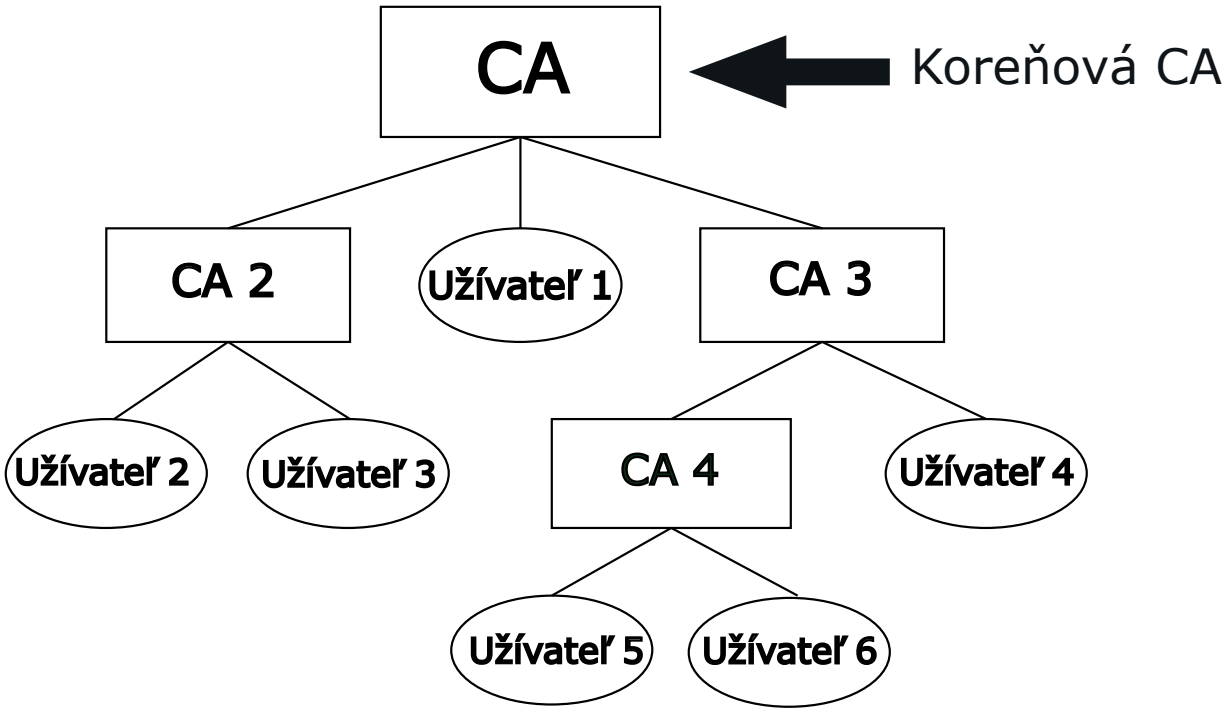
2.4.2 Uchovávanie informácií

CA udržuje databázu užívateľov, ktorá musí byť chránená, pretože obsahuje citlivé údaje o užívateľoch. CA môže prevádzkovať aj archív, ktorý obsahuje certifikáty, ktorých platnosť vypršala (táto povinnosť pre niektoré CA vyplýva zo zákona). Vytvára takisto zoznam vydaných certifikátov, ktorý býva voľne dostupný na verejnej stránke [2].

2.4.3 Zrušenie certifikátu

Pri vydávaní certifikátu sa očakáva, že bude platný po celú dobu platnosti, avšak certifikát môže byť zrušený[2] ešte pred jej vypršaním. Môže to byť spôsobené rôznymi okolnosťami ako napr. zmenou mena subjektu, odcudzením súkromného kľúča alebo ak zamestnancovi (držiteľovi certifikátu pre pracovné účely) skončí pracovný pomer [12]. Zrušené certifikáty sú publikované na zozname zrušených certifikátov

[2]Pojem zrušený je definovaný v slovenskej legislatíve, preto bude v preklade používaný v tejto podobe.



Obr. 2.3: Ukážka stromovej štruktúry certifikačných autorít

(CRL) až do konca ich pôvodnej platnosti. Aktualizované zoznamy CRL sú vydávané v určitých časových intervaloch. Táto skutočnosť môže byť nevyhovujúca napr. pri odcudzení súkromného kľúča užívateľa, preto môžu niektoré CA poskytovať on-line stav svojich certifikátov. Pre on-line zisťovanie stavu certifikátu slúži protokol OCSP (Online Certificate Status Protocol). OCSP však nenahrádza CRL, ale zrýchľuje spracovanie stavu certifikátu. [2]. OCSP klient vyšle požiadavku OCSP respondéru (serveru) na zistenie stavu certifikátu a pozdrží akceptovanie certifikátu minimálne do prijatia odpovede podpísanej OCSP respondérom. OCSP protokol definuje dáta, ktorých výmena (medzi aplikáciou a odpovedajúcim serverom) je potrebná pre zistenie stavu jedného alebo viac certifikátov. Ak OCSP respondér vie, že nastala kompromitácia súkromného kľúča CA, tak odpovede by mali obsahovať stav „zrušený“ pre všetky certifikáty vydané touto CA. Existuje základný typ OCSP odpovede, ktorý musí byť podporovaný všetkými OCSP servermi a klientmi. [13].

2.4.4 Obnovenie certifikátu

Ak sa blíži koniec doby platnosti certifikátu, je možné jeho platnosť obnoviť. Ak je daný certifikát stále platný, môže ho užívateľ využiť pre svoju autentifikáciu bez osobnej návštevy registračnej autority. Najjednoduchším spôsobom je elektronicky podpísať platným certifikátom žiadosť o nový certifikát vo forme správy. Obnovenie

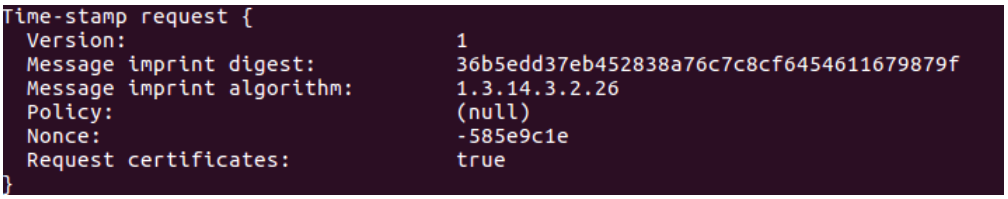
certifikátu môžu sprevádzať dva rôzne mechanizmy. Obnovenie certifikátu rovnakého verejného kľúča alebo s vygenerovaním nového páru kľúčov. Pri obnovení certifikátu rovnakého verejného kľúča sa budú líšiť len položky ako napr. sériové číslo certifikátu a doba platnosti. Pri obnovení s vygenerovaním nového páru kľúčov je nutné doložiť dôkaz o vlastníctve pôvodného a aj nového súkromného kľúča [2].

2.5 Autorita časových pečiatok

Autorita časových pečiatok alebo skrátene TSA (Timestamping Authority) je tvorená vydávajúcimi jednotkami TSU, ktoré majú vlastný pár kľúčov uložený napr. v HSM module. Na podpisovanie musí slúžiť súkromný kľúč používaný iba k tomuto účelu. Komunikácia je špecifikovaná protokolom pre vydávanie časových pečiatok TSP (Time Stamping Protocol) podľa štandardu RFC 3161. TSA neskúma totožnosť žiadateľa, ale autentifikácia žiadateľa pomocou certifikátu je výhodná aj z hľadiska bezpečnosti. Dôležitým faktorom je aj použitie dôveryhodného časového zdroja. Presnosť časového zdroja TSA je uvádzaná v politike pre vydávanie časových pečiatok. Aby bola zaručená vyššia presnosť, môže byť zdrojov času viac (ideálne 3 na sebe nezávislé zdroje) [2].

2.5.1 Vydávanie časových pečiatok

Vydávanie časových pečiatok prebieha pomocou autority časových pečiatok na základe žiadosti v štandardizovanom tvare. Túto žiadosť je možné vidieť na obr. 2.4. Obsahuje verziu protokolu TSP v položke „version“. Položka „message imprint“ definuje odtlačok samotného dokumentu a objetový identifikátor (OID) použitého algoritmu na tento odtlačok. TSA by mala podľa OID skontrolovať či sa jedná o prijateľný algoritmus z hľadiska bezpečnosti alebo či vyhovuje politike TSA. Potrebná je aj kontrola dĺžky odtlačku, aby zodpovedala použitému algoritmu. Voliteľná položka „policy“ môže obsahovať OID politiky, podľa ktorej by si klient prial vydanie časovej pečiatky. Položka „nonce“ obsahuje náhodné číslo dostatočnej veľkosti, ktoré bude prekopírované aj do výslednej odpovede. Nastavením položky „request certificate“ na hodnotu „true“ si žiadateľ praje, aby s časovým razítkom bol vrátený aj certifikát TSA (pre overenie digitálneho podpisu časovej pečiatky). Po odoslaní žiadosti prichádza odpoveď od TSA [2]. Jednotlivé položky odpovede TSA sú popísané v kap. 6.1.



Obr. 2.4: Žiadosť o časovú pečiatku

2.5.2 Platnosť časových pečiatok

Z pohľadu CA je TSA špeciálny prípad koncového užívateľa. Certifikát TSA má niekoľko odlišností od certifikátov bežných koncových užívateľov. Líši sa minimálne dĺžkou platnosti a použitím iba na verifikáciu časových pečiatok. Platnosť časovej pečiatky je obmedzená platnosťou certifikátu TSA [2].

3 MOŽNOSTI REALIZÁCIE VLASTNEJ PKI

Na internete je dostupných niekoľko softvérových riešení PKI a to úplne zdarma. Väčšina open-source PKI softvérov je založená na OpenSSL, nástrojovom balíku pre TLS a SSL protokoly. V tejto kapitole budú prezentované možnosti vytvorenia certifikačnej autority.

3.1 Programovanie vlastnej CA

Pre naprogramovanie certifikačnej autority je možné použiť viacero programovacích jazykov ako napríklad Java alebo Python. Výhodou programovania vlastnej CA je vytvorenie systému, ktorý plne zodpovedá našim predstavám či už zabezpečením, spôsobom vytvárania certifikátov alebo vizuálnou zložkou grafického rozhrania. Pre tieto potreby je možné použiť množstvo knižníc, ktoré obsahujú vytvorené šifrovacie funkcie a procedúry.

Nevýhodou sú nároky na programátorské zručnosti, pretože výsledkom by malo byť vytvorenie bezchybného systému, ktorý tvoria nielen jednoduché, ale aj programátorsky náročné súčasti. Preto je výhodnejšie použitie už existujúceho open-source softvéru.

3.2 OpenXPKI

Projekt založený na OpenSSL, ktorý poskytuje knižnice pre základné kryptografické funkcie. Tento projekt poskytuje webové rozhranie kompatibilné s najpoužívanejšími webovými prehliadačmi. Projekt zahŕňa aj ukážkovú konfiguráciu, na ktorej je možné vyskúšať činnosť softvéru. Je možné súbežne prevádzkovať viacero nezávislých CA. Pozitívna je taktiež možnosť použitia HSM (hardware security module). Takisto podporuje protokol SCEP (Simple Certificate Enrollment Protocol). Ďalšie informácie vrátane dokumentácie sa nachádzajú na oficiálnej stránke `http://www.openxpki.org/`.

3.3 XCA

Softvér obsahuje užívateľsky príjemné grafické rozhranie. Výhodou je možné použitie čipových kariet avšak najväčšou nevýhodou je celkovo zlé zabezpečenie napr. používanie staršej verzie TLS1.1 alebo využívanie hašovacej funkcie SHA-1, ktorá sa stala neakceptovateľná pre Microsoft, Google, Apple, Mozillu a pre mnoho ďalších od začiatku roku 2017, pretože už nie je dostatočne bezpečná [14]. Takisto nie je

možná konfigurácia on-line zisťovania stavu certifikátu. Informácie o tomto softvéri sú dostupné na stránke `https://hohnstaedt.de/xca/`.

3.4 PnP-CA

Celým názvom „Plug and play Certification authority“ je implementácia CA v jazyku Java s webovým rozhraním. Jedinou výhodou je ľahká konfigurácia (ako napovedá názov), no nie je garantované potrebné zabezpečenie a ani potrebná funkčnosť. Softvér je dostupný na stránke `https://sourceforge.net/projects/pnp-ca/`.

3.5 EJBCA

Open-source, ktorý je veľmi prispôsobiteľný. Podobne ako pri OpenXPKI je možný súbeh neobmedzeného množstva CA v rámci jednej inštancie. Na overenie či nebol certifikát zrušený je možné použiť zoznam zrušených certifikátov (CRL) alebo protokol OCSP. Ďalej ponúka výber viacerých šifrovacích algoritmov. Takisto je zaistená podpora väčšiny štandardných protokolov napr. SCEP, CMP. Používanie štandardu PKCS#11 umožňuje použitie HSM pre väčšiu ochranu. Informácie o softvéri vrátane dokumentácie sú dostupné na stránke `https://www.ejbca.org/`.

3.6 Porovnanie a výber najvhodnejšieho riešenia

Na základe porovnania vyššie uvedených open-source softvérov, ktoré je zhrnuté v tab. 3.1, boli zhodnotené ako najlepšie 2 softvéry EJBCA a OpenXPKI. S prihliadnutím na fakt, že EJBCA ponúka navyše možnosť doplnenia protokolu OCSP a neboli zistené vážnejšie nedostatky, tak pre diplomovú prácu bude najvýhodnejšie použitie tohto softvéru. Softvér ponúka svoju veľkú prispôsobiteľnosť, pekný vzhľad grafického rozhrania a potrebné funkcie. Navyše spoločnosť, ktorá vyvíja tento softvér ponúka volne dostupnú aplikáciu SignServer, ktorá slúži na vytváranie digitálnych podpisov a časových pečiatok.

3.6.1 EJBCA

Enterprise Java Beans Certificate Authority je softvér sponzorovaný švédskou spoločnosťou Primekey Solutions AB a jeho zdrojový kód je dostupný pod podmienkami GNU Lesser General Public License, avšak sú dve verzie softvéru: certifikovaná verzia **enterprise** a necertifikovaná verzia **community**. Keďže EJBCA využívajú aj veľké spoločnosti s vyššími nárokmi na zaistenie dôvery, bola vytvorená verzia enterprise, ktorá môže navyše zaisťovať ďalšie funkcie podľa potreby ako napr. ochranu

Tab. 3.1: Porovnanie open-source softvérov

Názov softvéru	Výhody / nevýhody
OpenXPKI	+ web UI + konfigurácia na vyskúšanie + súbeh viacerých nezávislých CA + podpora SCEP protokolu + možnosť použitia HSM + CRL – absencia OCSP
XCA	+ grafické rozhranie – nedostatočné zabezpečenie – TLS1.1 – SHA-1
PNP-CA	+ ľahká konfigurácia + webové rozhranie – nedostatočné zabezpečenie – nízka prispôsobitelnosť
EJBCA	+ vysoká prispôsobiteľnosť + súbeh viacerých nezávislých CA + CRL, OCSP + možnosť použitia HSM

integrity databázy a pre tieto firmy je takisto zabezpečená podpora zo strany vývojárov softvéru. Spoločnosti a užívatelia, ktorí nevyžadujú funkcie certifikovanej verzie, môžu využívať verziu community, do ktorej sú takisto postupne pridávané nové funkcie, ktoré boli dávnejšie pridané do verzie enterprise.

V práci sa bude ďalej pracovať s verziou community, ktorá poskytuje všetky funkcie, potrebné pre vytvorenie fungujúcej certifikačnej autority. Keďže sa jedná o open-source softvér, k dispozícii sú zdrojové kódy a konfiguračné súbory, ktoré vytvárajú veľkú prispôsobiteľnosť podľa predstáv užívateľa. Je umožnené aj použitie čipových kariet, USB tokenov alebo použitie HSM (hardware security module) pre bezpečné uchovanie kľúčov. Nevýhodou verzie community je, že nie je narozdiel od platenej verzie skúšaná priamo vývojármi na rôznych druhoch a verziách aplikačných serverov a operačných systémoch.

3.6.2 SignServer

SignServer je volne dostupná aplikácia od rovnakej spoločnosti Primekey Solutions AB. Podobne ako EJBCA je dostupný v dvoch edíciách enterprise a community. Slúži na podpisovanie dokumentov, ale takisto aj na vytváranie časových pečiatok. Keďže EJBCA túto funkciu neposkytuje, v diplomovej práci bude realizovaná práve pomocou SignServer-u edíciou community. Informácie o aplikácii sú dostupné na stránke `https://www.signserver.org/`.

4 INŠTALÁCIA VYBRANÝCH SOFTVÉROV

V rámci diplomovej práce bude vytvorená certifikačná autorita softvérom EJBCA a autorita časových pečiatok aplikáciou SignServer. CA bude realizovaná na kompletnej distribúcii operačného systému Linux s názvom Ubuntu 16.04.4 LTS dostupného z URL: `https://www.ubuntu.com/download/desktop` na školskom počítači v miestnosti SC5.34 pomocou softvéru VMware slúžiaceho pre virtualizáciu počítačov.

Pre začiatok je potrebné nainštalovať open-source implementáciu Java SE 7 s názvom OpenJDK-7 (Open Java Development Kit), na ktorú sa vzťahuje GNU GPL licencia. Je dostupná aj verzia OpenJDK-8, ale aplikačný server, ktorý bude spomenutý v ďalšej časti, nemá podporu pre platformu Java 8. Ďalej je potrebný aplikačný server, ktorý tvorí vrstvu medzi operačným systémom a aplikáciami. Bolo vyskúšaných viacero verzií aplikačných serverov od spoločnosti JBoss v súčinnosti so softvérom EJBCA a aplikačný server JBoss AS 7.1.1.Final bol zvolený ako najvhodnejší. Je to takisto open-source aktuálne vyvíjaný spoločnosťou Red Hat. Je dostupný na stránke `http://jbossas.jboss.org/downloads`. Pri tejto verzii je po jeho stiahnutí potrebné iba rozbalenie a spustenie pomocou príkazu *./standalone.sh* v domovskom adresári aplikačného serveru v zložke */bin*. Niekedy je nutné spustenie tohto súboru z domovského adresára pomocou úplnej cesty k súboru *./standalone.sh*, z dôvodu výskytu neočakávaných chýb pri spustení.

```
$ /home/user/jboss-as-7.1.1.Final/bin/standalone.sh
```

Softvér EJBCA je dostupný na stránkach `www.sourceforge.net/projects/ejbca/`. Inštalovaná bude verzia EJBCA 6.10.

Na zostavenie aplikácie bude použitý nástroj Apache Ant, ktorého princíp je podobný ako u unixového nástroja Make. Je to open-source projekt organizácie Apache Software Foundation a je používaný pri veľkom množstve projektov zásluhou svojej jednoduchosti a nezávislosti na platforme. Umožňuje kompiláciu, testovanie alebo vytvorenie balíku pre distribúciu [15].

4.1 Inštalácia EJBCA

Dokumentácia EJBCA je dostupná na oficiálnej stránke `https://www.ejbca.org/docs/EJBCA_Documentation.html`. Pred samotnou inštaláciou je potrebné oboznámenie sa s konfiguračnými súbormi EJBCA nachádzajúcich sa v priečinku *conf*. Tento priečinok obsahuje viac vzoriek konfiguračných súborov z ktorých sú najdôležitejšie súbory *ejbca.properties.sample, cesecore.properties.sample a install.properties.sample*.

Ak chceme zmeniť parametre CA je potrebné zmeniť názov súboru aby neobsahoval *.sample*.

V konfiguračnom súbore *ejbca.properties* je nutné definovať domovský adresár aplikačného serveru a aj to aký aplikačný server bude použitý. Takisto sa tu definujú rôzne parametre ako napr. parameter obsahujúci cestu k súboru, ktorý obsahuje informácie o údržbe serveru.

V súbore *cesecore.properties* sa definuje algoritmus pre generáciu náhodných čísel, ktorý vytvára sériového čísla certifikátov. Prednastaveným algoritmom je algoritmus „SHA1PRNG“. V jazyku Java je tento generátor pseudo-náhodných čísel poskytovaný triedou SecureRandom. Táto implementácia, ktorú obsahuje OpenJDK záhŕňa špecifikácie obsiahnuté v štandarde DSS (Digital Signature Standard). Keďže počítače sú deterministické zariadenia, nie je možné vygenerovať čisto náhodné čísla. Je však možné premeniť určité množstvo náhodne vygenerovaných bitov pomocou PRG (pseudorandom generator) na vygenerovanie veľkého množstva bitov, ktoré „vyzerajú“ ako náhodné [16]. Implementácia v jazyku Java vyzerá nasledovne:

```
SecureRandom random = SecureRandom.getInstance("SHA1PRNG");
```

Pri generovaní dlhých CRL záleží od množstva RAM prideleného aplikačnému serveru, koľko položiek môže byť z databázy vybraných za určitý čas, preto je možné definovať hodnotu, kde nižšia znamená menej položiek (viac opakovaní výberu). Takisto je tu definovaná dĺžka sériového čísla v bitoch, zoznam zakázaných znakov v databáze alebo definovanie primárneho jazyku. Pre lepšiu ochranu proti „off-line“ útokom hrubou silou na prelomenie hesiel, môže byť použitý výpočtovo náročný algoritmus BCrypt. Nastavenie hodnoty od 1 do 31 určuje mieru bezpečnosti, kde vyššie číslo znamená nárast výpočtovej náročnosti. BCrypt je hašovacia funkcia navrhnutá Nielsom Provosom a Davidom Mazieresom, založená na symetrickej blokovej šifre Blowfish. BCrypt dosahuje odolnosť voči útokom pomocou dúhových tabuliek, pretože v sebe zahŕňa kryptografickú soľ s dostatočnou veľkosťou 128 bitov a jedná sa o adaptívnu funkciu. To umožňuje ľuďom zvýšiť počet iterácií a spomaliť tak výpočet. Bez ohľadu na zvolený počet iterácií, tento počet by mal byť z času na čas zmenený. V OpenBSD implementácii BCrypt hašované heslá začínajú s prefixom $2a$, $2b$ alebo $2y$ [21]. V práci bude zvolená hodnota počtu iterácií 6, pretože postačuje potrebnému zabezpečeniu. V tomto súbore je možné takisto vylúčiť niektoré podporované šifrovacie sady, aby neboli použité z dôvodu bezpečnosti napr. použitie klasickej Diffie-Hellmanovej výmeny kľúčov je menej bezpečné než sa zopár rokov dozadu zdalo [18].

Ďalším hlavným konfiguračným súborom je *install.properties*. Už z názvu je jasné, že slúži na definovanie parametrov k inštalácii. Mnohé z položiek je však možné zmeniť aj pri samotnej inštalácii ako napr. názov CA, jedinečné meno CA

atď.

V súbore sa definuje aj spôsob vytvárania kľúčov CA, kde je možné použiť aj HSM, ale v tejto realizácii bude použitá možnosť softvérovo generovaných kľúčov pomocou algoritmu RSA s dĺžkou kľúča 2048 bitov, ktorého bezpečnosť sa zatiaľ považuje za dostatočnú. Algoritmus RSA bol prvýkrát verejne popísaný v roku 1978 a jeho názov tvoria začiatočné písmená priezvisk autorov (Rivest, Shamir a Adleman). Je to algoritmus, ktorý je vhodný pre podpisovanie aj pre šifrovanie [22]. Narozdiel od kryptografie eliptických kriviek (ECC), ktoré sú v posledných rokoch čoraz populárnejšie, algoritmus RSA bol populárny už od svojho vzniku. Patrí medzi asymetrické kryptosystémy a je založený na zložitosti faktorizácie (rozloženia na súčin prvočísiel) veľkých čísiel [20]. Nevýhodou RSA je, že na dosiahnutie rovnakej úrovne bezpečnosti ako pri ECC je potrebné použitie dlhšieho kľúča. Napr. dĺžka kľúča 3072 bitov pri RSA zodpovedá rovnakej úrovni bezpečnosti ako dĺžka kľúča 256 bitov pri ECC [19].

Na podpisovanie bude slúžiť dvojica algoritmov, ktorú bude tvoriť hašovacia funkcia SHA-256, keďže funkcia SHA-1 už nie je dostatočne bezpečná proti dobre vybaveným útočníkom a podpisový algoritmus RSA (dĺžka kľúča 2048 bitov) [14]. SHA-256 patrí pod súhrnné označenie SHA-2 spolu s ďalšími funkciami, ktoré sa líšia číslom určujúcim dĺžku výsledného odtlačku (SHA-224, SHA-256, SHA-384, SHA-512) [8]. Platnosť certifikátu CA s názvom KorenovaCA, ktorý bude podpísaný samotnou CA, bola stanovená na 10 rokov. Ak by vytváraná CA vydávala certifikáty reálnym zákazníkom, je nutné, aby sa jej názov nezhodoval so žiadnou inou existujúcou CA.

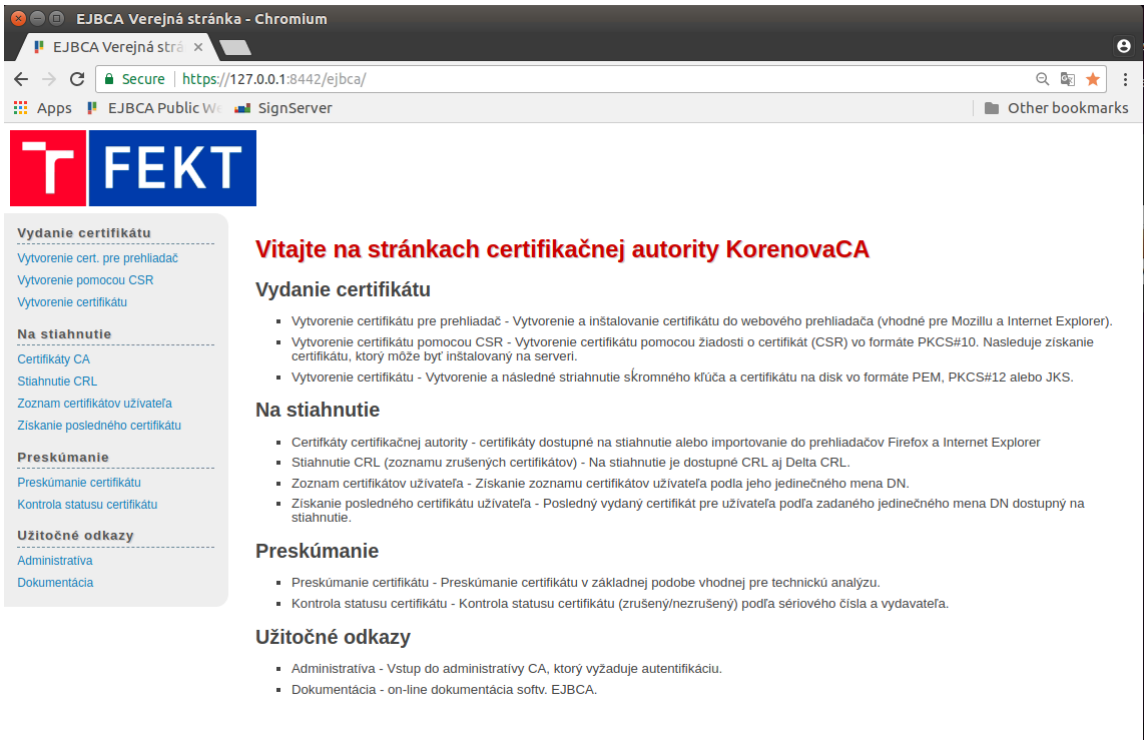
V práci bude použitá preddefinovaná HSQLDB (Hyper SQL databáza), ale bolo vyskúšané aj použitie databázového systému MariaDB po stiahnutí ovládača a konfigurácioe v príkazovom riadku aplikačného serveru. Pri používaní EJBCA vo väčších firmách je odporúčané použitie iného databázového systému než HSQLDB, pretože časom tento systém zaberá väčšie množstvo pamäte pri potrebe vydania veľkého množstva certifikátov.

Kompilácia sa inicializuje príkazom *ant deploy* v domovskom adresári EJBCA (musí byť spustený aj aplikačný server). Tento príkaz slúži na rozvinutie potrebných nastavení aj do aplikačného serveru a pred dokončením je nutné aj nastavenie hesla databázy. Príkaz *ant install* je potrebný na spustenie inštalácie. Táto inštalácia vygeneruje potrebné certifikáty, kľúče a ďalšie náležitosti potrebné pre funkčnosť koreňovej CA. Po reštartovaní aplikačného serveru je dostupná admin stránka, do ktorej má prístup len držiteľ certifikátu, ktorý bol vygenerovaný v zložke */p12* domovského adresára EJBCA a následne importovaný do internetového prehliadača pod názvom *Superadmin*. Do internetového prehliadača je takisto potrebné importovať certifikát CA. Admin stránka vytvorenej certifikačnej autority s názvom **KorenovaCA**, je

dostupná na adrese `https://127.0.0.1:8443/ejbca/adminweb/` (localhost).

4.2 Úprava verejnej stránky

Inštaláciou sa vytvorí určitá štruktúra verejnej stránky, ktorú je možné prispôsobiť podľa vlastných potrieb a poskytovaných služieb. Technológia JavaServer Pages (JSP) pomáha obsluhovať dynamicky generované webové stránky založené napr. na HTML. Na modifikáciu bolo použité vývojové prostredie Eclipse IDE. Upravenú verziu stránky je možné vidieť na obr. 4.1. Verejná stránka slúži užívateľom ako poskytovateľ služieb, pre ktoré nie je nutné mať administrátorské práva. Ak užívateľ obdržal prihlasovacie údaje, môže vytvoriť určitý typ certifikátu. Ďalej je na verejnej stránke dostupný zoznam zrušených certifikátov (CRL), certifikáty koreňovej CA alebo je možné vyhľadať certifikáty vydané užívateľom.



Obr. 4.1: Grafické rozhranie verejnej stránky CA

4.3 Inštalácia SignServer

Na realizáciu autority časových pečiatok bude využitý softvér SignServer. Je takisto volne dostupný na stránke `https://sourceforge.net/projects/signserver/`.

V diplomovej práci bude využitá community verzia SignServer 4.0.0-bin. Nastavenie parametrov pred inštaláciou prebieha v konfiguračnom súbore *signserver_deploy.properties* v zložke */signserver-ce-4.0.0/conf*. Inštalácia sa spustí príkazom *bin/ant deploy* z domovského adresára SignServeru. Po inštalácii je možné vytvárať tzv. worker-ov, ktorí majú na starosti jednotlivé funkcie napr. podpisovanie PDF dokumentov, časové pečiatky. Toto je možné docieliť upravovaním pripravených súborov v zložke *doc/sample-configs* a následným prídaním „worker-a“. Community verzia SignServer ponúka demo verziu webovej stránky, na ktorej je možné otestovať inštaláciu a vytvorených „worker-ov“ na adrese: `https://127.0.0.1:8442/signserver/`. Správu SignServeru je možné vykonávať pomocou príkazového riadku alebo pomocou grafického rozhrania, ktoré je možné spustiť príkazom:

```
$ bin/signserver-gui
```

5 TESTOVANIE SLUŽIEB CERTIFIKAČNEJ AUTORITY

Pre každú CA je mimoriadne dôležitá ochrana súkromného kľúča a takisto aj jeho záloha, ktorú je vhodné vytvoriť. Základ tejto ochrany tvorí uchovanie súkromného kľúča na zariadení HSM. Dôležitým faktorom je aj fyzická ochrana, ktorá má zabrániť odcudzeniu zariadení a pri manipulácii s týmito zariadeniami je nutné dodržovanie množstva bezpečnostných opatrení. Keďže CA realizovaná v tejto diplomovej práci nebude vydávať certifikáty reálnym zákazníkom, použitie takýchto bezpečnostných prvkov nie je nutné, avšak bezpečnosť pripojenia bude zabezpečená.

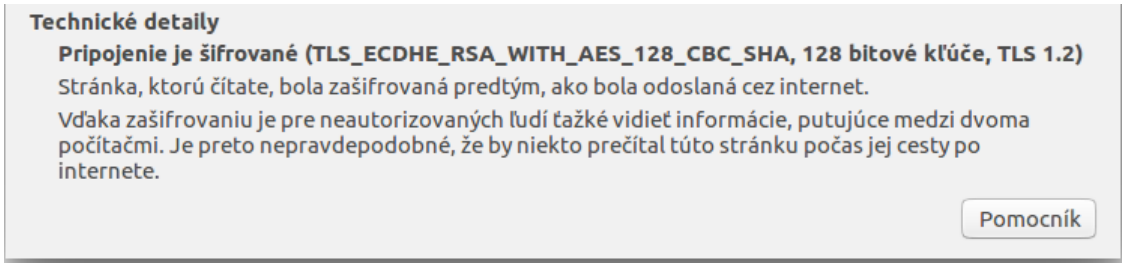
5.1 Zabezpečená komunikácia

Transport layer security (TLS) je kryptografický protokol, ktorý poskytuje zabezpečenú komunikáciu na internete. Jeho predchodcom bol protokol SSL (Secure sockets layer). Protokol TLS umožňuje aplikáciám klient/server komunikovať tak, aby bolo možné sa vyvarovať napr. odpočúvaniu alebo falšovaniu prenášaných dát. Hlavným cieľom TLS protokolu je poskytovať dôvernosť a integritu dát dvom komunikujúcim aplikáciám. TLS protokol pozostáva z viacerých menších protokolov, no najhlavnejšie sú dva: zahajovací protokol a prenosový protokol. Prenosový protokol (record protocol), ktorý poskytuje zabezpečenie spojenia má dve základné vlastnosti. Spojenie je dôverné. Na šifrovanie dát sa používa symetrické šifrovanie (napr. AES). Kľúče potrebné pre šifrovanie sú generované pomocou vopred dohodnutého tajomstva pre každé spojenie a to zabezpečuje ich jedinečnosť. Druhou vlastnosťou je dôveryhodnosť, kedy je možné overiť integritu dát pomocou hašovacej funkcie. Zahajovací protokol (handshake protocol) má na starosti umožnenie vzájomnej autentifikácie serveru a klienta (niekedy je vyžadovaná len autentifikácia serveru), dohodnutie šifrovacieho algoritmu a ustanovenie kľúčov pred spustením prenosového protokolu. Autentifikácia je prevažne založená na základe verejných kľúčov, ktoré sú získané z digitálnych certifikátov a vlastníctva príslušných súkromných kľúčov jednotlivými stranami [23].

Vytvorením certifikátu pri inštalácii pre server je možné realizovanie zabezpečeného pripojenia pomocou TLS. Aktuálna osvedčená verzia protokolu TLS je TLS 1.2, ale existuje aj finálna podoba internetového konceptu TLS 1.3, ktorý sa v blízkej dobe môže stať internetovým štandardom. TLS 1.2 ponúka široký výber šifrovacích sád. Šifrovacia sada je súbor kryptografických algoritmov a vyjadruje spôsob autentifikácie, ustanovenia symetrických kľúčov, algoritmus šifrovania a typ hašovacej

funkcie. Pre TLS 1.2 je definovaná aj sada, ktorej podpora je povinná pre obe komunikujúce strany a jej podoba je TLS_RSA_WITH_AES_128_CBC_SHA [23].

Na obr. 5.1 je vidieť verzia protokolu (TLS 1.2) a dohodnutá šifrovacia sada pri komunikácii so serverom. ECDHE_RSA označuje algoritmus ustanovenia kľúčov, tvorený protokolom Diffie-Hellman s využitím eliptických kriviek, kde certifikát serveru musí byť podpísaný algoritmom RSA a vlastniť verejný kľúč vhodný pre RSA. Šifrovanie dát prebieha použitím 128-bitovej symetrickej šifry AES v režime CBC. Typ použitej hašovacej funkcie je SHA. Bezpečnosť, ktorú táto šifrovacia sada poskytuje je zatiaľ všeobecne považovaná za dostačujúcu a je stále podporovaná všetkými internetovými prehliadačmi. Vstup na verejnú stránku pre vydanie certifikátu je možný aj cez nezabezpečené spojenie použitím portu 8080, ale nutnosťou je poskytnutie zabezpečeného spojenia pomocou https protokolu použitím portu 8442, ktorý narozdiel od použitia portu 8443 nevyžaduje obojstrannú autentifikáciu, ale iba autentifikáciu serveru pomocou certifikátu. Toto nastavenie prebieha na aplikačnom serveri v súbore *standalone.xml* príkazom *verify-client="true"* ako je možné vidieť na obr. 5.2.



Obr. 5.1: Použitá šifrovacia sada komunikácie so serverom

```
            <connector name="httpspriv" protocol="HTTP/1.1" scheme="https" socket-
binding="httpspriv" secure="true">
                <ssl key-alias="localhost" password="serverpwd" certificate-key-
file="/home/user/jboss-as-7.1.1.Final/standalone/configuration/keystore/
keystore.jks" verify-client="true" ca-certificate-file="/home/user/jboss-
as-7.1.1.Final/standalone/configuration/keystore/truststore.jks" ca-certificate-
password="heslo"/>
            </connector>
```

Obr. 5.2: Nastavenie obojstrannej autentifikácie

5.2 Vytvorenie certifikátu

Pomocou koreňovej CA je možné vydávať certifikáty rôznym užívateľom alebo ďalším certifikačným autoritám a vytvárať tak stromovú štruktúru CA. V tejto časti

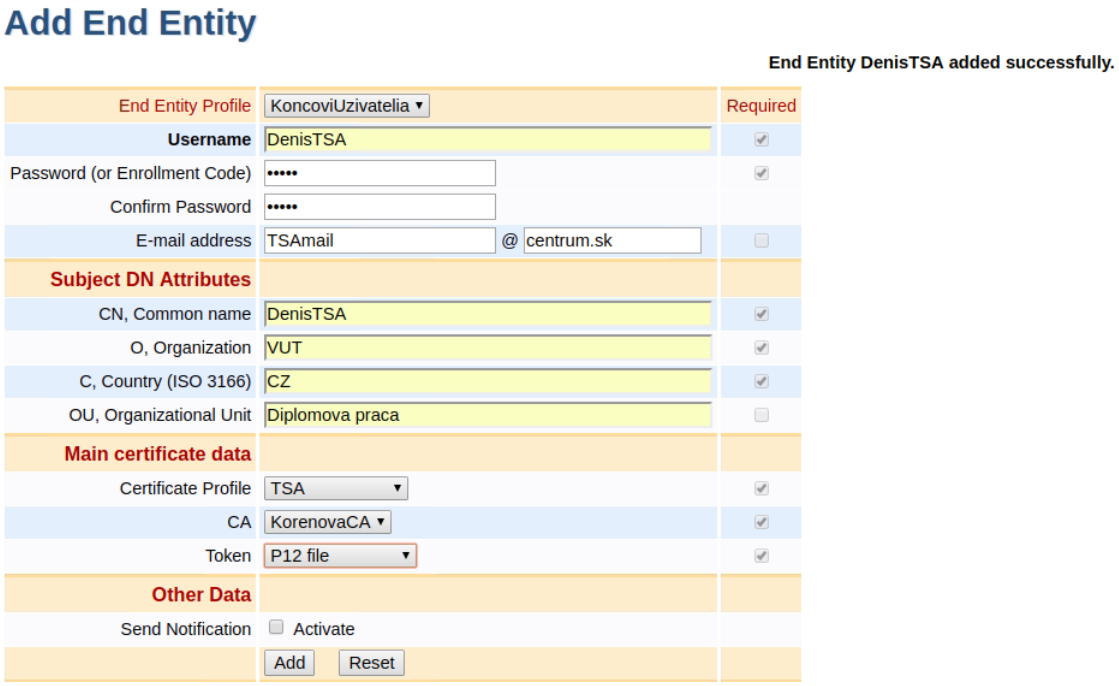
budú vytvorené certifikáty potrebné pre vytvorenie elektronického podpisu a časových pečiatok. Vytvorenie certifikátov je možné aj pomocou príkazového riadku, ale z dôvodu nutnej špecifikácie množstva parametrov veľmi nepraktické. Pre tieto účely je vhodné využitie admin stránky. V záložke *Certificate Profiles* je možné vytáranie nových certifikačných profilov a definovanie jednotlivých parametrov.

5.2.1 Certifikačné profily

Pre potreby vydania certifikátu a vygenerovania dvojice kľúčov autorite časových pečiatok (TSA podľa RFC3161), bol vytvorený profil TSA. V tomto profile sa definujú dostupné algoritmy na generáciu kľúčov (RSA, DSA) a ich bitová dĺžka. Ďalej je možné definovať podpisový algoritmus, ktorý však záleží od CA, ktorá certifikát vydáva. Keďže TSA certifikáty mávajú zväčša dlhšiu dobu platnosti ako certifikáty bežných užívateľov, bola platnosť certifikátu stanovená na 5 rokov. Základné obmedzenia, ktoré obsahujú napr. informáciu či užívateľ certifikátu bude koncová entita alebo certifikačná autorita boli označené ako závažné (critical). Medzi použitie kľúča bolo zvolené použitie na digitálny podpis (označené ako závažné). Certifikát určený pre TSA musí obsahovať iba jednu položku rozšíreného použitia kľúča označenú ako závažnú, ktorou je časové pečiatkovanie (timestamping) [17]. Ďalšou položkou je definovanie certifikačnej politiky. Certifikačná politika je definovaná objektovým identifikátorom OID. Pri koncových entitách OID určuje, politiku pod ktorou bol daný certifikát vydaný a na aké účely by mal byť použitý. Aplikácie so špeciálnymi požiadavkami by mali mať zoznamy OID akceptovateľných politík a porovnať ich s OID uvedených v rozšírení. [12]. Certifikát takisto obsahuje URL distribučného miesta CRL v položke lokalita distribúcie CRL a URL pre skontrolovanie on-line stavu certifikátu v položke prístup k informáciám autority.

Po zadefinovaní všetkých potrebných parametrov je možné prejsť k vytvoreniu samotnej koncovej entity. U kvalifikovaných poskytovateľov služieb vytvárajúcich dôveru prebieha tento proces vydávania certifikátov vytvorením žiadosti a následným navštívením registračnej autority žiadateľom, ktorý musí doniesť potrebné dokumenty potvrdzujúce jeho identitu napr. občiansky preukaz. Ak je žiadateľom právnická osoba, okrem preukazu totožnosti je potrebné potvrdenie o existencii obchodnej spoločnosti a dokument, ktorý oprávňuje danú osobu jednať za spoločnosť. Pri vytváraní koncovej entity je nutné, aby mal držiteľ certifikátu jedinečné (unikátne) meno v rámci celej CA. Nutnosťou je definovanie prihlasovacieho mena a hesla, ktoré bude distribuované žiadateľovi pre prihlásenie do systému a vygenerovaniu kľúčového páru a vytvoreniu certifikátu. Ďalej je potrebný výber vopred vytvoreného certifikačného profilu a definovanie CA, ktorá bude mať úlohu vydavateľa (v prípade ak je vytvorených viacero CA). Poslednou definovanou položkou je formát vytvo-

reného súboru, v ktorom budú uchovaný súkromný kľúč a certifikát. Na obr. 5.3 je možné vidieť tento proces vytvárania koncovej entity. Samotné prihlásenie a generovanie je možné pomocou verejnej stránky výberom algoritmu príslušnej bitovej dĺžky. Následne sú certifikát a príslušný súkromný kľúč uložené podla PKCS (public-key cryptography standards) štandardov PKCS#12 v súbore s príponou *.p12*. Vytvorený certifikát autority časových pečiatok s názvom DenisTSA je na obr. 5.4.



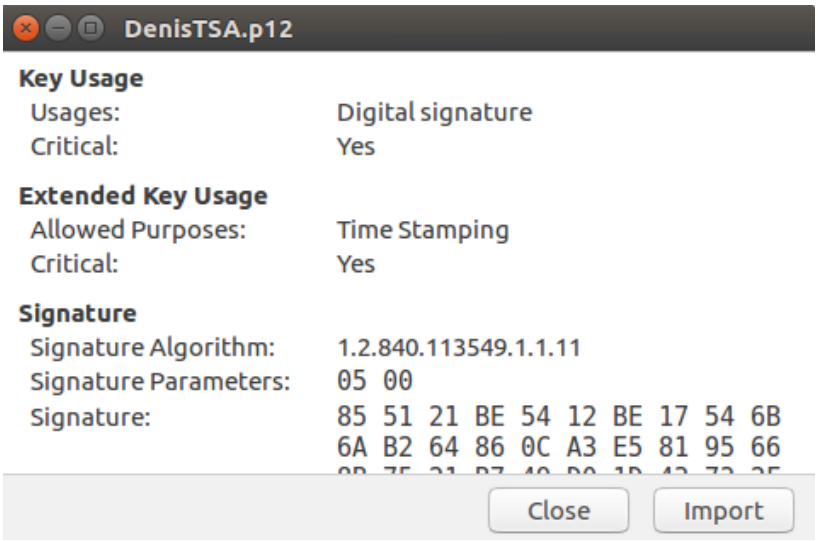
Obr. 5.3: Vytváranie koncovej entity

Obdobne je potrebné vytvoriť certifikát vhodný pre elektronické podpisovanie. Hlavný rozdiel je pri definovaní použitia kľúča na elektronický podpis. Tento certifikát bude vydaný užívateľovi Denis. Vhodné je oddeliť dvojice kľúčov používané na autentifikáciu a elektronické podpisovanie z dôvodu bezpečnosti. Pri autentifikácii nie je zrejmé aká správa alebo aký reťazec je podpisovaný a týmto spôsobom by mohol vzniknúť podpis nechcených dát.

5.3 Zoznam zrušených certifikátov

Zoznam zrušených certifikátov[1] (CRL) je vydávaný certifikačnou autoritou alebo entitou poverenou na vydávanie tohto zoznamu. Tvoria ho certifikáty, ktorých stav

[1]Tento pojem je používaný v slovenskej legislatíve, preto bude preklad CRL uvádzaný v tejto podobe.

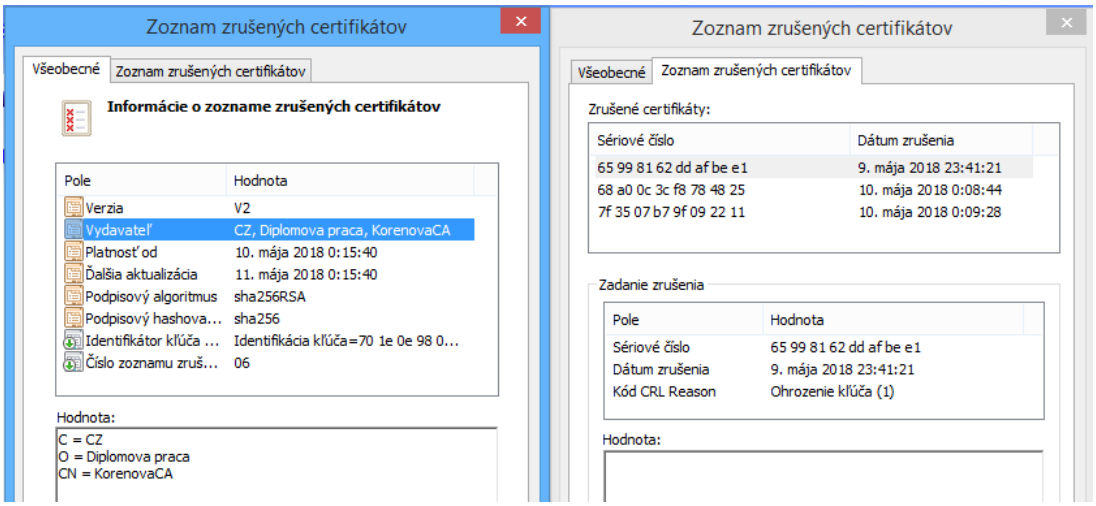


Obr. 5.4: Certifikát TSA

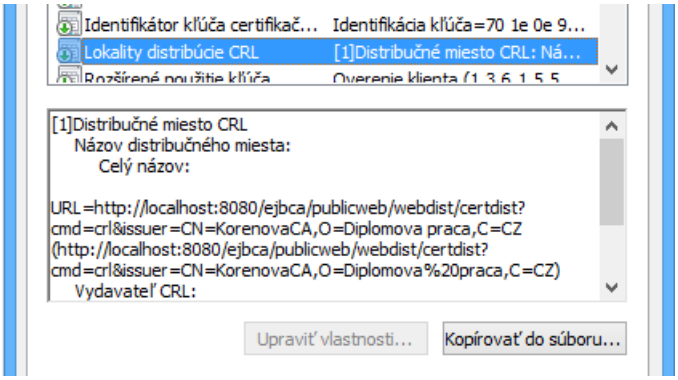
bol na žiadosť užívateľov alebo aj za iných okolností (napr. zrušením CA), zmenený na „zrušený“ (revoked). Dôvodov môže byť hneď niekoľko napr. strata alebo odcudzenie súkromného kľúča, zmena dôležitých údajov, koniec pracovného pomeru. Ak už je certifikát zrušený, tento stav je pri mnohých CA trvalý a certifikát nie je možné obnoviť.

EJBCA umožňuje pridať službu automatického generovania CRL v určitom časovom intervale napr. každých 24 hodín. Všetky vygenerované CRL sú uchovávané v databáze až do manuálneho vymazania. Takisto je možné vydávať tzv. delta CRL, ktorý obsahuje certifikáty, ktorých stav sa zmenil od vydania kompletného CRL. Na obr. 5.5 je možné vidieť CRL vydaný vytvorenou certifikačnou autoritou KorenovaCA, ktorý je dostupný na verejnej stránke (súbor *KorenovaCA.crl*). Ako bolo spomínané, distribučné miesto CRL by malo byť uvádzané v rozšíreniach vydaných certifikátov. Toto rozšírenie obsahuje URL adresu distribučného miesta, pomocou ktorého je automaticky stiahnutý súbor s príponou *.crl*. Distribučné miesto bolo pridávané do vydaných certifikátov (obr. 5.6).

CRL obsahuje sériové čísla zrušených certifikátov, dátum ich zrušenia a mnohokrát aj dôvod. Medzi informáciami o CRL sa nachádza napr. verzia (verzia 2 ak CRL obsahuje rozšírenia), vydavateľ CRL a algoritmus podpísania vydaného zoznamu. Položka „platnosť od“ obsahuje dátum vydania CRL a položka „ďalšia aktualizácia“ obsahuje dátum ďalšieho vydania (reálny dátum ďalšieho vydania CRL môže byť skorší) [12].



Obr. 5.5: Zoznam zrušených certifikátov



Obr. 5.6: Distribučné miesto uvedené vo vydanom certifikáte

5.4 Zisťovanie stavu certifikátu online

V praxi je z bezpečnostného hľadiska vyžadované oddelenie služby on-line overovania stavu certifikátu od certifikačnej autority, aby CA nemusela akceptovať žiadne ďalšie prichádzajúce žiadosti alebo z dôvodu dostupnosti počas údržby CA. V diplomovej práci slúži CA zároveň aj ako OCSP respondér.

On-line overovanie stavu certifikátu bolo vyskúšané na aktívnom ale aj na odvolanom certifikáte. Na overenie stavu je možné využitie openSSL, kedy je potrebné definovať certifikát CA, certifikát overovaného užívateľa a URL OCSP respondéra. Následne je vytvorená žiadosť, na ktorú odpovedá OCSP respondér podpísanou odpoveďou. Hlavné položky odpovede OCSP respondéra je možné vidieť na obr. 5.7. Všetky OCSP odpovede by mali byť elektronicky podpísané a minimálne obsahovať stav daného certifikátu (v tomto prípade je stav certifikátu „revoked“). Odpoveď

obsahuje napr. verziu syntaxu odpovede (verzia 1), haš verejného kľúča respondéra, čas vygenerovania odpovede, sériové číslo certifikátu, ktorého stav sa zisťuje a podpisový algoritmus. Pri stave „zrušený“ (revoked) nasledujú položky čas zrušenia a dôvod zrušenia. Výhodou využitia OCSP je, že zmena stavu certifikátu sa prejaví okamžite a nie je potrebné čakať na ďalšie vydanie ako pri CRL.

```
OCSP Response Data:
    OCSP Response Status: successful (0x0)
    Response Type: Basic OCSP Response
    Version: 1 (0x0)
    Responder Id: 701E0E98068C84F8DEBF2D3FDD6A64643E21D845
    Produced At: May 15 19:35:38 2018 GMT
    Responses:
    Certificate ID:
      Hash Algorithm: sha1
      Issuer Name Hash: 4A7F9EB9A6DE1F6395EEE8BD35F995AD2CF0E335
      Issuer Key Hash: 701E0E98068C84F8DEBF2D3FDD6A64643E21D845
      Serial Number: 0F6CEC54CEE8D517
    Cert Status: revoked
    Revocation Time: May 15 19:34:34 2018 GMT
    Revocation Reason: keyCompromise (0x1)
    This Update: May 15 19:35:38 2018 GMT

    Response Extensions:
        OCSP Nonce:
            0410CFF211DA22031EAAB2CED03D28FF77B1
    Signature Algorithm: sha1WithRSAEncryption
         e3:51:3f:84:f2:ee:d1:4e:a1:33:c6:71:ab:a9:0e:59:68:c8:
         bb:04:6f:ae:95:fb:0b:87:d7:ae:6b:1c:06:c1:9e:b7:7d:b1:
         17:36:99:f3:f9:57:29:07:c5:65:9f:05:ca:b3:74:52:de:ac:
```

Obr. 5.7: Podpísaná odpoveď OCSP respondéru

5.5 Obnovenie certifikátu

Obnovenie certifikátu v praxi znamená vydanie nového certifikátu s rovnakým verejným kľúčom, aký bol uvedený v predošlom certifikáte. To znamená, že sériové číslo nezostáva rovnaké ako u predošlého certifikátu.

6 VYTVORENIE AUTORITY ČASOVÝCH PEČIATOK

Ako bolo spomenuté, SignServer slúži na vytvorenie autority časových pečiatok a digitálne podpisovanie. Funguje na princípe „worker-ov“, ktorí majú určité funkcie. Každý worker potrebuje mať pre správne fungovanie priradeného určitého „crypto worker-a“, ktorý obsahuje cestu k súkromnému kľúču a certifikátu. Pre vytvorenie crypto worker-a slúži súbor *keystore-crypto.properties* v zložke *doc/sample-configs*. V tomto súbore je nutné zmeniť cestu k vytvorenému súkromnému kľúču a certifikátu (*KEYSTOREPATH*), v tomto prípade DenisTSA. Pri použití súboru vo formáte PKCS#12 (prípona *.p12*) je potreba nastaviť *KEYSTORETYPE* na PKCS12. Nutnosťou je takisto uvedenie hesla, ktoré bolo zadávané pri vytváraní certifikátu (položka *KEYSTOREPASSWORD*). V položke *DEFAULTKEY* je potrebné uviesť užívateľské meno (prípadne CN) predošle vytvorenej entity. Následne je možné vytvoriť tohto crypto worker-a pomocou príkazového riadku. Vytvorenie mu priradí určité ID, ktoré je potrebné pre ďalšiu konfiguráciu. Pre vytvorenie autority časových pečiatok (TSA) podľa RFC3161 slúži súbor *timestamp.properties*. Pre správne fungovanie je nutnosť definovať názov správneho crypto worker-a v položke *CRYPTOTOKEN*. Možná je takisto definícia politiky TSA. Položka *ACCURACYMILLIS* určuje kvalitu časového zdroja stanovením odchýlky. Je možné definovať podporované algoritmy a vylúčiť napr. algoritmus SHA-1. Podpisový algoritmus *SIGNATUREALGORITHM* bol stanovený na SHA256withRSA. TSA musí mať aj interný zdroj času. SignServer má implementovanú triedu na použitie času počítaču ako dôveryhodný časový zdroj. Pri vydávaní časových pečiatok reálnym zákazníkom by takýto zdroj času nebol prijateľný, ale na demonštráciu funkcie TSA je dostačujúci. TSA majú v ideálnom prípade viac dôveryhodných časových zdrojov (3 zaručené zdroje) [2].

6.1 Vytvorenie časovej pečiatky

Vytvorenie časovej pečiatky spočíva vo vytvorení žiadosti, ktorá je následne poslaná TSA. Následne je vygenerovaná odpoveď od TSA. Výpis odpovede TSA na základe príkazu -outrep je možné vidieť na obr. 6.1. Tento výpis, ktorý bol vygenerovaný ako odpoveď na žiadosť klienta, obsahuje hodnotu 0 v položke „status“, ktorá značí, že všetko prebehlo v poriadku. Jadrom časového razítka je štruktúra TSTInfo, ktorá je digitálne podpísaná TSA síce efektívne ale neprehľadným spôsobom. Položka „gen time“ obsahuje čas vydania pečiatky (používa sa formát UTC) a „gen time accuracy“ vyjadruje presnosť tohto času. V položke „message imprint“ je uvedený od-

tlačok, ktorý bol časovo opečiatkovaný a skopírovaný zo žiadosti o časovú pečiatku. Položka „nonce“ je takisto skopírovaná zo žiadosti a slúži na logické spojenie žiadosti a odpovede. „Serial number“ obsahuje poradové číslo časovej pečiatky, ktoré musí byť v rámci TSA jedinečné. Položka „policy“ obsahuje OID politiky TSA, ktorá vydala časovú pečiatku [2].

```
user@ubuntu:~/signserver-ce-4.0.0$ bin/signclient timestamp -print -inrep /home/user/Desktop/odpoved10.tsr
Time-stamp response {
  Status:                        0
  Status message:                Operation Okay
  Time-stamp token:
      Info:
         Accuracy:                org.bouncycastle.asn1.tsp.Accuracy@10
         Gen Time:                Tue May 08 14:12:03 GMT+01:00 2018
         Gen Time Accuracy:       1.000000
         Message imprint digest:  36b5edd37eb452838a76c7c8cf6454611679879f
         Message imprint algorithm: 1.3.14.3.2.26
         Nonce:                   20f11eb9
         Serial Number:           3c88f55bf24a5921
         TSA:                     (null)
         Policy:                  1.3.6.1.4.1.22408.1.2.3.45
      Signer ID:
         Serial Number:           451d06a155912125
         Issuer:                  CN=KorenovaCA,O=Diplomova praca,C=CZ
      Signer certificate:
         Certificate:
            Serial Number:        451d06a155912125
            Subject:              CN=DenisTSA,OU=Diplomova praca,O=VUT,C=CZ
            Issuer:               CN=KorenovaCA,O=Diplomova praca,C=CZ
      Other certificates:
         Certificate:
            Serial Number:        4279d7f521333759
            Subject:              CN=KorenovaCA,O=Diplomova praca,C=CZ
            Issuer:               CN=KorenovaCA,O=Diplomova praca,C=CZ
}
2018-05-08 15:07:52,900 INFO  [TimeStampCommand] Processing took 224 ms
```

Obr. 6.1: Výpis odpovede autority časových pečiatok

6.1.1 Overenie časovej pečiatky

Keďže časová pečiatka je dátová štruktúra digitálne podpísaná TSA, overenie sa realizuje pomocou certifikátu TSA. Keďže certifikát TSA vydala nejaká certifikačná autorita, je treba overiť či sa daný certifikát nenachádza napr. na zozname zrušených certifikátov alebo zistiť či nebol zrušený pomocou OCSP. Nasleduje spočítanie odtlačku pôvodnej správy alebo dokumentu a porovnanie s odtlačkom uvedeným v časovom razítku [2]. Overenie digitálneho podpisu pomocou certifikátu TSA je na obr. 6.2.

```
[TimeStampCommand] Token was validated successfully.
[TimeStampCommand] Token was generated on: Tue May 08 14:12:03 GMT+01:00 2018
[TimeStampCommand] MessageDigest=36b5edd37eb452838a76c7c8cf6454611679879f
[TimeStampCommand] Processing took 167 ms
```

Obr. 6.2: Overenie časovej pečiatky

6.2 Digitálne podpisovanie PDF

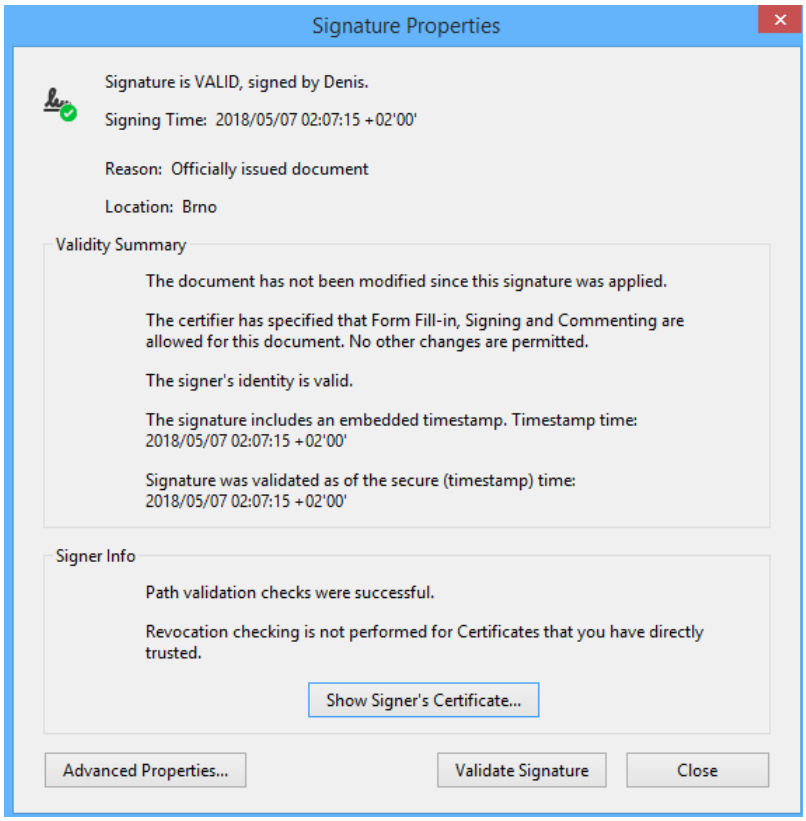
Aby bolo možné pomocou SignServeru elektronicky podpisovať PDF súbory, je potreba vytvoriť nového crypto workera rovnako ako predošlého, ktorý bude obsahovať cestu k certifikátu vytvorenému na elektronické podpisovanie s názvom Denis. Aby bolo možné rozlíšiť crypto worker-ov je potrebné novému dať iný názov a ten potom priradiť k worker-ovi na PDF podpisovanie. Toho je možné vytvoriť pomocou súboru *pdfsigner.properties*. V položke *DIGESTALGORITHM* je možné vybrať ako hašovaciu funkciu SHA-256, ktorá bude použitá aj pri podpise. Pri DSA kľúčoch je možné použiť iba algoritmus SHA-1. Ponúkaná je aj možnosť časovej pečiatky elektronického podpisu pridaním názvu worker-a TimeStampSigner v položke *TSA_WORKER*, ktorá bude použitá. Aktivácia prebehne rovnako ako v predchádzajúcich prípadoch so zmeneným ID. Pre kontrolu aktivácie všetkých potrebných worker-ov je možné použiť príkaz:

```
$ bin/signserver getstatus complete all
```

Podpisovanie prebieha na strane serveru nahraním PDF súboru na webovej stránke. Následne prebehne vytvorenenie a stiahnutie elektronicky podpísaného a opečiatkovaného PDF súboru. Ak bol certifikát podpisujúceho sa užívateľa zrušený, tak časová pečiatka pri elektronickom podpise slúži aj pre informáciu, či bol dokument podpísaný pred alebo po uvedení certifikátu na CRL.

6.2.1 Overenie digitálneho podpisu a časovej pečiatky

Program Adobe Acrobat Reader DC umožňuje overenie elektronického podpisu a časovej pečiatky. Ako aj internetové prehliadače, tak aj programy ako Adobe Acrobat Reader obsahujú prednastavený zoznam dôveryhodných certifikačných autorít alebo certifikátov. Overením autenticity sa zistí, či sa certifikát alebo nadradené certifikáty užívateľa, ktorý vytvoril podpis, nachádzajú na zozname dôveryhodných. Platnosť podpisového certifikátu sa určí podľa konfigurácie aplikácie Adobe Acrobat Reader. Nasleduje overenie integrity dokumentu, či neprišlo k zmene obsahu od aplikovania podpisu. Ak bol obsah dokumentu zmenený, program určí či to bolo povoleným spôsobom (môže byť povolené pridávanie komentárov). Overovanie časových pečiatok takisto vyžaduje získanie certifikátu TSA do zoznamu dôveryhodných certifikátov. Ako je možné vidieť na obr. 6.3 časová pečiatka aj elektronický podpis sú po overení označené ako platné.



Obr. 6.3: Detaily digitálneho podpisu s časovou pečiatkou

7 LABORATÓRNA ÚLOHA

Nasledujúca kapitola obsahuje vytvorenie laboratórnej úlohy pre oboznámenie sa s infraštruktúrou verejných kľúčov. Pre čo najvhodnejší výber spôsobu inštalácie pre laboratórnu úlohu boli vyskúšané viaceré typy na viacerých aplikačných serveroch. Pri vybraných verziách softvérov je možné použitie spoločného aplikačného serveru narozdiel od niektorých starších verzií.

7.1 Infraštruktúra verejných kľúčov

Cieľom laboratórnej úlohy je zoznámenie sa s infraštruktúrou verejných kľúčov, certifikačnou autoritou a autoritou časových pečiatok. V rámci úlohy bude vytvorená certifikačná autorita, certifikát autority časových pečiatok a bude vyskúšaná funkčnosť vytvorenej autority časových pečiatok.

7.1.1 Príprava pracoviska a inštalácia

Laboratórna úloha bude realizovaná na kompletnej distribúcii operačného systému Linux s názvom Ubuntu 16.04.4 LTS pomocou softvéru VMware. Softvér na realizovanie certifikačnej autority EJBCA je dostupný na stránke `https://www.ejbca.org/download.html` alebo na `https://sourceforge.net/projects/ejbca/files/ejbca6/`, kde je možné vybrať konkrétnu verziu. V laboratórnej úlohe sa bude pracovať s verziou ejbca_ce_6_10_1_2.zip, preto je potrebné jej stiahnutie. Aby nebolo nutné riešiť problémy pri inštalácii, ktoré sa týkajú práv užívateľov a pod., tak je vhodné rozbaliť stiahnutý .zip súbor do domovského adresára. Pre skrátenie času inštalácie bude vykonaná zjednodušená inštalácia pomocou skriptu v zložke bin/extra/ejbca-setup. Je potreba nainštalovať implementáciu Java 8 nástro Apache ant a potrebné súčasti pre inštaláciu databázového systému MariaDB.

```
$ sudo apt-get update
$ sudo apt-get install unzip openjdk-8-jdk-headless ant ant-optional
   ↪ psmisc mariadb-client bc patch curl
```

Inštalácia mariaDB

```
$ sudo apt-get update -y
$ sudo apt-get install mariadb-server
```

A následne je potrebné vytvoriť databázu ejbcatest (s právami užívateľa ejbca a heslom ejbca) príkazmi:

```
$ sudo mysql -u root -p
```

```
mysql> CREATE DATABASE ejbcatest CHARACTER SET utf8 COLLATE
   ↪ utf8_general_ci;
mysql> GRANT ALL PRIVILEGES ON ejbcatest.* TO 'ejbca'@'localhost'
   ↪ IDENTIFIED BY 'ejbca';
```

Následne by malo byť všetko pripravené na inštaláciu. Je možné pozrieť obsah skriptu /bin/extra/ejbca-setup.sh a skontrolovať údaje, prípadne zmeniť názov databázy ak bola vytvorená databáza s iným názvom. Spustením skriptu ejbca-setup.sh z domovského adresára (pretože skript vytvorí nové priečinky) sa začne inštalácia. Je potreba odpovedať na výzvy trikrát číslicou 1 (znamená yes):

```
$ ./ejbca_ce_6_10_1_2/bin/extra/ejbca-setup.sh
```

Po úspešnom nainštalovaní je vypísaná zložka, v ktorej sa nachádza certifikát pre prístup k administratívnej stránke a heslo k tomuto certifikátu, ktoré bude použité pri importovaní do internetového prehliadača Mozilla. Pre import je nutné otvoriť v prehliadači Menu > Preferences > Security > Advanced > Certificates > View Certificates a v položke „your certificates“ pomocou možnosti „Import“ nahrať certifikát superadmin.p12 zo zložky /ejbca_ce_6_10_1_2/p12 a zadať heslo, ktoré bolo vypísané po inštalácii. Následne na verejnej stránke na adrese `localhost:8080/ejbca` pomocou možnosti Fetch CA Certificates v menu je možné sa dostať k certifikátu CA a importovať ho do prehliadača ako dôveryhodnú certifikačnú autoritu možnosťou „Download to Firefox“. Následne je však ešte potrebné ísť do nastavení, kde bol importovaný certifikát do prehliadača a v položke „Authorities“ vyhľadať certifikát CA (Example CA - Management CA) a pomocou nastavenia Edit Trust povoliť príslušné možnosti. Týmto je možné zabezpečiť bezpečnú komunikáciu medzi serverom a klientom (webová stránka a internetový prehliadač) pomocou protokolu TLS. Digitálne certifikáty slúžia v tomto prípade na autentifikáciu komunikujúcich strán a sú základom ustanovenia bezpečnej komunikácie.

Po týchto nastaveniach je možné nainštalovať softvér SignServer pre vytvorenie časovej pečiatky. Dokumentáciu je možné nájsť na stránke `https://www.signserver.org`. Na stránke `https://sourceforge.net/projects/signserver/files/signserver/4.0/` je možné stiahnutie konkrétnej verzie signserver-ce-4.0.0-bin.zip. Takisto je vhodnejšie rozbalenie v domovskej zložke /home/user. Pre SignServer je potrebné nastavenie systémovej premennej *APPSRV_HOME*. Toto nastavenie je možné pomocou súboru */etc/environment*, do ktorého je potrebné vložiť s administrátorskými právami *APPSRV_HOME="/home/user/wildfly-10.1.0.Final"* a pre vyhnutie sa varovaniam v logu je nutné takisto vložiť jedinečné ID serveru napr. *SIGNSERVER_NODEID =node1*. Načítanie systémových premenných je docielené príkazom:

```
$ source /etc/environment
```

Pred samotnou inštaláciou je potrebné nastavenie určitých parametrov. Nastavenie prebieha v konfiguračnom súbore, ktorý bude vytvorený skopírovaním a premenovaním súboru *signserver_deploy.properties.sample* na *signserver_deploy.properties* v zložke */signserver-ce-4.0.0/conf*. Je potrebné súbor otvoriť pomocou textového editoru a odkomentovať možnosť *database.name=nodb*, vytvoriť zložku *nodb* s právom zapisovania, odkomentovať a upraviť cestu k tejto zložke napr.

```
database.nodb.location="/home/user/wildfly-10.1.0.Final/standalone/
    ↪ data/nodb"
```

Je vhodné odkomentovať *appserver.home* a definovať celú cestu k aplikačnému serveru, aby pri inštalácii bola detekovaná verzia.

```
appserver.home=/home/user/wildfly-10.1.0.Final
```

Ostatné preddefinované parametre by mali zaistiť správnu funkčnosť. Na opravenie problému s webovou službou je potrebné otvoriť príkazový riadok aplikačného serveru.

```
$ /home/user/wildfly-10.1.0.Final/bin/jboss-cli.sh
```

Príkazom *connect* sa pripojiť k serveru a nakonfigurovať ho postupne nasledujúcimi príkazmi (po každom príkaze počkať na výpis outcome => "success"):

```
/subsystem=webservices:write-attribute(name=wsdl-host, value=jbossws.
    ↪ undefined.host)
/subsystem=webservices:write-attribute(name=modify-wsdl-address,
    ↪ value=true)
:reload
```

Inštalácia sa spustí príkazom *bin/ant deploy* z domovského adresára SignServeru. Výpis úspešnej inštalácie je na obr. 7.1.

```
     [echo] Deploying /home/user/signserver-ce-4.0.0/tmp/signserver.ear
/user/wildfly-10.1.0.Final/standalone/deployments
     [echo]
     [echo] Task completed 2018-05-20 15:52:51 -0700.

BUILD SUCCESSFUL
Total time: 5 seconds
user@ubuntu:~/signserver-ce-4.0.0$
```

Obr. 7.1: Úspešná inštalácia SignServer.

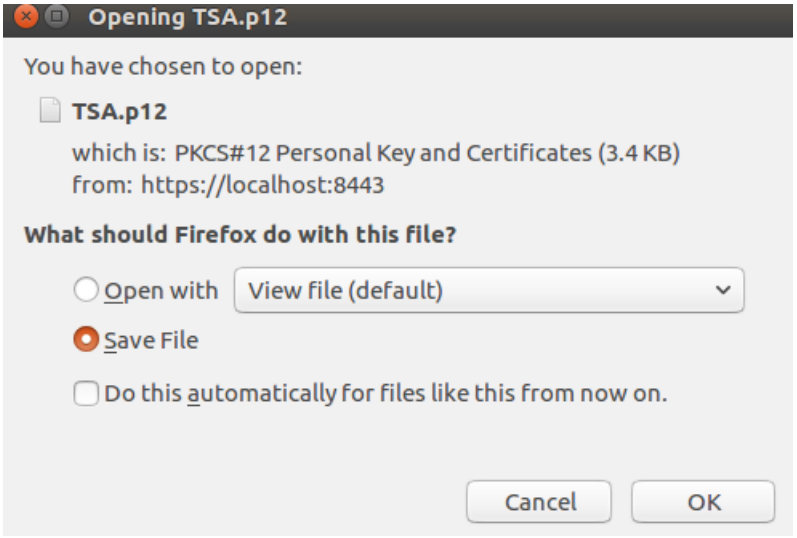
7.1.2 Vytvorenie certifikátu pre TSA

Po úspešnej inštalácii je možné pomocou certifikačnej autority vytváranie a správa certifikátov. Certifikát slúži ako dôkaz, že daná osoba, ktorej bol vydaný je vlastníkom odpovedajúceho súkromného kľúča k verejnému kľúču, ktorý je uvedený v certifikáte. Preto je možné použiť certifikát TSA na overenie podpisu, ktorým podpisuje TSA odpoveď obsahujúcu časovú pečiatku.

Vytvorenie certifikátu pre autoritu časových pečiatok je možné pomocou admin stránky `https://localhost:8443/ejbca/adminweb`. Najprv je potrebné vytvorenie certifikačného profilu v záložke „Certificate Profiles“ v hlavnom menu. Je potreba pridať certifikačný profil napr. s názvom TSA tlačítkom „add“. Následne je možné tento profil editovať. Typ zostane End Entity. Takisto položky dostupných algoritmov a ich bitovej dĺžky môžu zostať nezmenené. Doba platnosti certifikátu TSA býva zväčša dlhšia ako u normálnych certifikátov, preto je možné ju zmeniť napr. na 5 rokov (5y). Pri položke „key usage“ ktorá je označená ako „critical“ zostane označené digitálne podpisovanie „digital signature“ a „non-repudiation“, ale je potrebné odznačiť „key encipherment“, ktoré označuje použitie na šifrovanie a prenos kľúča. „Extended key usage“ je rozšírené použitie kľúča. Pri autoritách časových pečiatok musí byť táto položka označená ako „critical“ a obsahovať iba jednu možnosť a to „Time stamping“. Ostatné položky zostanú nezmenené a takto nakonfigurovaný profil je možné uložiť. Po uložení je možné prejsť k vytvoreniu profilu koncovej entity „End entity profiles“ a ak nie je vytvorený profil TSA, je nutné ho pridať a následne editovať. V položke „Subject DN Attributes“ je možné vybrať a pridať „O, Organization“ a „C, Country“. Je treba pridať certifikačný profil TSA v položke„Available Certificate Profiles“ a „Default token“ nastaviť na „P12 file“. Takto definovaný profil je možné uložiť. Posledným krokom pred vytvorením certifikátu je vytvorenie koncovej entity záložkou „Add end entity“. V „End entity profile“ je potrebné vybrať vytvorený profil TSA. Položky „username“ a „password“ slúžia neskôr pre autentifikáciu užívateľa pri vytváraní certifikátu a heslo slúži aj pre prístup k vytvorenému úložisku certifikátu a súkromného kľúča, preto je potrebné si ho zapamätať. Je nutné stanoviť jedinečné meno (DN), ktoré sa skladá z položiek CN (napr. TSA), organizácia O (napr. VUT) a krajina C (napr. CZ) (je vyžadované minimálne stanovenie CN). V položke „Certificate“ profile je takisto potrebné vybrať vytvorený profil TSA. Ostatné položky zostanú nezmenené a je možné pridať užívateľa.

Generovanie úložiska certifikátu a súkromného kľúča prebieha cez verejnú stránku `https://localhost:8443/ejbca` v záložke „Create keystore“. Je potrebné zadať definované meno a heslo a následne zvoliť algoritmus generovania kľúčov (napr. RSA 2048 bits). Malo by začať sťahovanie certifikátu a súkromného kľúča v súbore s prí-

ponou .p12 podľa štandardu PKCS#12 (obr. 7.2). Je výhodné vytvoriť priečinok,



Obr. 7.2: Stiahnutie certifikátu a súkromného kľúča.

do ktorého bude daný súbor uložený, pretože je potrebné aj stiahnutie certifikátu vo formáte PEM (.pem). Tento certifikát je možné nájsť pomocou záložky „Search end entities“ v hlavnom menu admin stránky a následnom vyhľadaní koncovej entity TSA. Je potrebné kliknúť na odkaz „View Certifikates“ koncovej entity TSA a následne stiahnuť pomocou „Download PEM file“ v ľavom rohu do rovnakej zložky ako TSA.p12.

7.1.3 Vytvorenie autority časových pečiatok

K vytvoreniu autority časových pečiatok bude slúžiť program SignServer, ktorý je možné spravovať buď pomocou príkazového riadku alebo pomocou grafického rozhrania, ktoré je možné spustiť pomocou príkazu

```
$ bin/signserver-gui
```

Prevažne však bude používaný príkazový riadok. Pre začiatok je potrebné nakonfigurovať „crypto token“, pomocou ktorého bude realizovaný prístup k súkromnému kľúču, bez ktorého by TSA nemohla podpisovať odpovede na žiadosti. Pre vytvorenie „crypto tokenu“ s názvom CryptoTokenP12 je nutné v konfiguračnom súbore doc/sample-configs/keystore-crypto.properties nakonfigurovať typ úložiska (PKCS12), cestu k nemu, heslo pre prístup do úložiska (zadávané pri vytváraní certifikátu) a názov kľúča. Konfigurácia vyzerá nasledovne (pozor na odkomentovanie príkazu):

```
# Type of keystore
# PKCS12 and JKS for file-based keystores
# INTERNAL to use a keystore stored in the database (tied to the
    ↪ crypto worker)
WORKERGENID1.KEYSTORETYPE=PKCS12
#WORKERGENID1.KEYSTORETYPE=JKS
#WORKERGENID1.KEYSTORETYPE=INTERNAL

# Path to the keystore file (only used for PKCS12 and JKS)
WORKERGENID1.KEYSTOREPATH=/home/user/TSAcert/TSA.p12

# Optional password of the keystore. If specified the token is "
    ↪ auto-activated".
#WORKERGENID1.KEYSTOREPASSWORD=heslo

# Optional key to test activation with. If not specified the first
    ↪ key found is
# used.
#WORKERGENID1.DEFAULTKEY=TSA
```

Konfiguráciu je možné uplatniť pomocou príkazov:

```
$ bin/signserver setproperties doc/sample-configs/keystore-crypto.
    ↪ properties
$ bin/signserver reload 1
```

Ak sa vyskytol problém s definovaním APPSRV_HOME, je potrebné definovanie pomocou príkazu:

```
$ export APPSRV_HOME=/home/user/wildfly-10.1.0.Final/
```

Ako je zrejmé, každý „worker“ má priradené určité ID, ktoré slúži na identifikáciu pri vykonávaní príkazov. Pri chybne zadanom údaji je možné „workera“ uvedenými príkazmi odstrániť a nakonfigurovať ešte raz (worker, ktorého ID je 1).

```
$ bin/signserver removeworker 1
$ bin/signserver reload all
```

Následne je potrebná konfigurácia súboru definujúceho základné parametre TSA. Konfigurácia prebieha v súbore doc/sample-configs/timestamp.properties. V tomto súbore je potrebné nakonfigurovať názov kľúča:

```
WORKERGENID1.DEFAULTKEY=TSA
```

Keďže nie je k dispozícii iný interný dôveryhodný zdroj času, je nutné použitie preddefinované zdroja času, ktorým je lokálny čas počítaču. Je možné definovať podpisový algoritmus a odkomentovať položku pre získanie názvu TSA z certifikátu:

```
WORKERGENID1.SIGNATUREALGORITHM=SHA256withRSA
WORKERGENID1.TSA_FROM_CERT=true
```

Vytvorenie opäť prebieha pomocou príkazov:

```
$ bin/signserver setproperties doc/sample-configs/timestamp.
   ↪ properties
$ bin/signserver reload 2
```

Ak niektorá položka chýba alebo bola zle nakonfigurovaná je možné ju doplniť/upraviť pomocou príkazu a následným načítaním pomocou príkazu reload (definovanie položky DEFAULTKEY):

```
$ bin/signserver setproperty (ID workera) DEFAULTKEY "TSA"
$ bin/signserver reload all
```

naopak vymazanie prebieha pomocou príkazu

```
$ bin/signserver removeproperty (ID workera) DEFAULTKEY
$ bin/signserver reload all
```

Pri správnej konfigurácii je Worker aj Token status v stave Active (obr. 7.3). Skontrolovanie stavu je možné pomocou príkazu:

```
$ bin/signserver getstatus complete all
```

```
Status of CryptoWorker with id 1 (CryptoTokenP12) is:
   Worker status : Active
   Token status  : Active

   Worker properties:
      KEYSTORETYPE=PKCS12

      DEFAULTKEY=TSA

      KEYSTOREPATH=/home/user/TSAcert/TSA.p12

      KEYSTOREPASSWORD=heslo

      NAME=CryptoTokenP12
```

Obr. 7.3: Správne nakonfigurovaný CryptoToken.

7.1.4 Vytvorenie časovej pečiatky

Pre vyskúšanie správnej funkčnosti a opečiatkovania nejakého dokumentu je potrebné vytvorenie napr. textového dokumentu s ľubovolným názvom a obsahom (napr. dokument.txt). Pre vytvorenie časovej pečiatky je potrebné využitie nasledujúcich príkazov:

```
$ bin/signclient timestamp -infile /home/user/Desktop/dokument.txt -
   ↪ outreq /home/user/Desktop/ziadost.tsq -certreq -outrep /home/
   ↪ user/Desktop/odpoved.tsr -url http://localhost:8080/signserver
   ↪ /tsa?workerName=TimeStampSigner
```

Za príkazom -infile nasleduje celá cesta k dokumentu, ktorý chceme označiť časovou pečiatkou. Príkaz -outreq zabezpečí uloženie žiadosti (je potrebné definovať adresár uloženia), ktorá obsahuje niekoľko položiek vrátane verzie protokolu TSP (Time-Stamp Protocol), hašu dokumentu a objektový identifikátor algoritmu akým haš vznikol (v community verzii je možné iba použitie algoritmu SHA-1). Príkazom -certreq je zabezpečené požadovanie certifikátu TSA. Príkaz -outrep zabezpečí uloženie odpovede TSA a za ním je takisto potrebné definovať adresár uloženia. Poslednou definovanou položkou je URL adresa vytvorenej TSA. Pri správnom vytvorení časovej pečiatky by mal byť vypísaný „status code“ s hodnotou 0 ako na obr. 7.4. Výpis položiek žiadosti alebo odpovede TSA je možné zobraziť pomocou

```
INFO  [TimeStampCommand] Got reply after 6551 ms
INFO  [TimeStampCommand] TimeStampRequest validated with status code: 0

INFO  [TimeStampCommand] Processing took 6982 ms
```

Obr. 7.4: Výpis pri správne vytvorenej časovej pečiatke.

príkazov:

```
$ bin/signclient timestamp -print -inreq /home/user/Desktop/ziadost.
   ↪ tsq
$ bin/signclient timestamp -print -inrep /home/user/Desktop/odpoved.
   ↪ tsr
```

Odpoveď TSA je digitálne podpísaná a obsahuje časový údaj (UTC formát), haš, ktorý je rovnaký ako v žiadosti, sériové číslo a napr. údaj o presnoti časového zdroja. Pred overením správnosti tohto podpisu je nutné využiť funkcie OpenSSL a skonvertovať odpoveď, aby bola kódovaná formátom base64. To je možné nasledujúcim príkazom:

```
$ openssl enc -in /home/user/Desktop/odpoved.tsr -out /home/user/
   ↪ Desktop/odpoved.tsr.b64 -a
```

Následne je možné overenie správnosti podpisu TSA. To je vykonané pomocou príkazu:

```
$ bin/signclient timestamp -verify -signerfile /home/user/TSAcert/TSA
   ↪ .pem -inrep /home/user/Desktop/odpoved.tsr.b64
```

kde signerfile je cesta k úložisku certifikátu TSA vo formáte .pem. Výpisom takto overeného podpisu by mal byť čas vytvorenia časovej pečiatky a haš pôvodného súboru. Aj keď overenie digitálneho podpisu dokáže, že digitálny podpis vytvorila naozaj TSA, je možné že tento certifikát bol zrušený (odvolaný) ešte pred koncom jeho platnosti. V praxi na toto overenie slúži zoznam zrušených certifikátov, ktorý zodpovedná certifikačná autorita pravidelne zverejňuje alebo online zisťovanie stavu certifikátu, ktoré reflektuje okamžitú zmenu platnosti certifikátu (protokol OCSP). Dôvodom odvolania certifikátu môže byť napr. kompromitácia (strata) súkromného kľúča. Ak nastane neočakávaná udalosť a to kompromitácia súkromného kľúča certifikačnej autority, tak nasleduje zrušenie (odvolanie) všetkých certifikátov vydaných touto certifikačnou autoritou. Odvolanie certifikátu vykonáva príslušná certifikačná autorita najčastejšie na žiadosť klienta, ktorý vlastníctvo certifikátu potvrdzuje napr. znalosťou hesla pre odvolanie certifikátu.

7.1.5 Otázky k laboratórnej úlohe

- Na čo slúži digitálny certifikát?
- Čo znamenajú skratky CN, O a C v položke certifikátu jedinečné meno (DN)?
- Aké môžu byť dôvody zrušenia (odvolania) certifikátu?
- Aké položky obsahuje žiadosť o časovú pečiatku?
- Aké položky obsahuje odpoveď autority časových pečiatok?
- Akým spôsobom je možné skontrolovať platnosť certifikátu?

8 ZÁVER

Prvým cieľom diplomovej práce bolo naštudovanie služieb, ktoré zaisťuje certifikačná autorita, autorita časových pečiatok a infraštruktúra verejných kľúčov. Predtým však bolo potrebné oboznámiť sa so zmenami, ktoré nastali v oblasti PKI po tom ako nadobudlo účinnosť nariadenie Európskeho parlamentu a Rady EÚ známe aj pod názvom eIDAS, ktoré viedlo k prijatiu nových zákonov v Českej republike a ostatných členských štátoch EÚ.

V druhej časti práce sú popísané služby, ktoré zaisťuje certifikačná autorita a autorita časových pečiatok a súčasti PKI ako elektronický podpis a elektronická pečať, ktoré rozlišujeme podľa niekoľkých úrovní na základe dôveryhodnosti. V tejto kapitole je bližšie popísaná štruktúra digitálneho certifikátu a význam jednotlivých položiek. Vydávanie a podpisovanie týchto certifikátov je úlohou certifikačnej autority. Jej dôveryhodnosť závisí na viacerých faktoroch a pri rozhodovaní, ktorej certifikačnej autorite dôverovať nám pomáha aj dokument s názvom politika certifikačnej autority.

Ďalšia kapitola obsahuje dôležité porovnanie open-source softvérov, pomocou ktorých je možné realizovať vlastnú PKI. Jednotlivé softvéry boli vyskúšané a porovnávané podľa funkcií, ktoré poskytujú užívateľovi, ale aj z hľadiska prispôsobiteľnosti. Najlepšie z tohto porovnania vyšli open-source softvéry s názvom EJBCA a SignServer, ktoré boli vybrané pre realizáciu PKI v diplomovej práci.

V piatej kapitole bola inštaláciou vytvorená certifikačná autorita podľa potrieb diplomovej práce. Takisto bola upravená webová stránka CA spôsobom, ktorý uľahčí orientáciu a poskytnutie potrebných služieb užívateľovi.

V hlavnej časti práce bola demonštrovaná funkčnosť služieb certifikačnej autority, popísaných v druhej časti práce, medzi ktoré patrí vytvorenie certifikátu, zrušenie certifikátu a následné vytvorenie zoznamu zrušených certifikátov alebo overovanie on-line stavu certifikátov pomocou OCSP protokolu. Následne bola vytvorená autorita časových pečiatok, ktorá využíva certifikát vytvorený v predošlej časti práce. Pomocou vytvorenej TSA a softvéru SignServer bola demonštrovaná funkčnosť aj ostatných služieb vytvorením časovej pečiatky a digitálneho podpisu s následným overením v programe Adobe Acrobat Reader.

Posledná kapitola obsahuje vytvorenú laboratórnu úlohu, ktorá bude slúžiť študentom k ľahšiemu pochopeniu a vytvoreniu vlastnej infraštruktúry verejných kľúčov pomocou open-source softvéru.

LITERATÚRA

[1] *NAŘÍZENÍ EVROPSKÉHO PARLAMENTU A RADY (EU) č. 910/2014 ze dne 23. července 2014 o elektronické identifikaci a službách vytvářejících důvěru pro elektronické transakce na vnitřním trhu a o zrušení směrnice 1999/93/ES.* Evropská unie, 1998-2015. [online]. 42 s. Dostupné z URL: <http://eur-lex.europa.eu/legal-content/CS/TXT/PDF/?uri=CELEX:32014R0910&from=EN>.

[2] DOSTÁLEK, L., VOHNOUTOVÁ, M. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. Computer Press, a. s.*, 2006, 534 s. ISBN 80-251-0828-7.

[3] *Zákon č. 297/2016 Sb. Zákonu o službách vytvářejících důvěru pro elektronické transakce.* Sbírka zákonů České republiky. 2016, částka 115, s. 4466-4472. ISSN 1211-1244.

[4] *Zákon č. 250/2017 Sb. Zákon o elektronické identifikaci* Sbírka zákonů České republiky. 2016, částka 89, s. 2719-2727. ISSN 1211-1244.

[5] NOVÁKOVÁ L., MV ČR. *Senát schválil zákon o elektronické identifikaci.* ředitelka odboru tisku a public relations MV ČR, 19.7.2017. Dostupné z URL: <http://www.mvcr.cz/clanek/senat-schvalil-zakon-o-elektronicke-identifikaci.aspx>.

[6] MV ČR. *Seznam kvalifikovaných poskytovatelů služeb vytvářejících důvěru a poskytovaných kvalifikovaných služeb vytvářejících důvěru.* Odbor eGovernmentu, 1.9.2017. Dostupné z URL: <http://www.mvcr.cz/clanek/seznam-kvalifikovanych-poskytovatelu-sluzeb-vytvarejicich-duveru-a-poskytovanych-kvalifikovanych-sluzeb-vytvarejicich-duveru.aspx>.

[7] ADAMS, C., LLOYD, S. *Understanding PKI: concepts, standards, and deployment considerations. Addison-Wesley Professional*, 2003.

[8] STALLINGS, W. *Cryptography and network security principles and practices, Fourth edition. Prentice Hall.*, 2005, 592 s. ISBN 0-13-187316-4.

[9] THOMSEN, S. S., KNUDSEN, L. R. *Cryptographic hash functions. Technical University of Denmark*, 2009.

[10] DOSEDĚL, T. *Počítačová bezpečnost a ochrana dat. Computer press*, 2004.

[11] DE NORMALISATION, COMITÉ EUROPÉEN, and EUROPÄISCHES KOMITEE FÜR NORMUNG. *Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures-Part 1: System Security Requirements.* In CEN Workshop Agreement CWA-14167-1. URL: <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/eSign/cwa14167-01-2003-Jun.pdf>.

[12] COOPER, D., SANTESSON, S., FARRELL, S., BOEYEN, S., HOUSLEY, R., POLK, W. *RFC 5280. Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* The Internet Engineering Task Force, 2008.

[13] SANTESSON, S., MYERS, M., ANKNEY, R., MALPANI, A., GALPERIN, S., ADAMS, C. *RFC 6960. X. 509 Internet public key infrastructure online certificate status protocol-OCSP.* The Internet Engineering Task Force, 2013.

[14] STEVENS, M., BURSZTEIN, E., KARPMAN, P., ALBERTINI, A., and MARKOV, Y. *The first collision for full SHA-1. Annual International Cryptology Conference*, s. 570-596, Springer, Cham, 2017.

[15] LOUGHRAN, S., HATCHER, E. *Ant in Action: of Java Development with Ant. Dreamtech Press*, 2007.

[16] BELLARE, M., ROGAWAY, P. *Introduction to modern cryptography. Ucsd Cse*, 2005.

[17] ADAMS, C., PINKAS, D. *RFC 3161. Internet X. 509 public key infrastructure time stamp protocol (TSP).* The Internet Engineering Task Force, 2001.

[18] ADRIAN, D., BHARGAVAN, K., DURUMERIC, Z., GAUDRY, P., GREEN, M., HALDERMAN, J.A., HENINGER, N., SPRINGALL, D., THOMÉ, E., VALENTA, L. and VANDERSLOOT, B. *Imperfect forward secrecy: How Diffie-Hellman fails in practice. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, s. 5-17, ACM, 2015.

[19] BARKER, E., BARKER, W., BURR, W., POLK, W., and SMID, M. *Recommendation for key management part 1: General (revision 3). NIST special publication 800*, s. 1-147, 2012.

[20] MAHTO, D., KHAN, D. A., and YADAV, D. K. *Security Analysis of Elliptic Curve Cryptography and RSA. Proceedings of the World Congress on Engineering.*, s. 419-422, 2016.

[21] PROVOS, N., MAZIERES, D. *A Future-Adaptable Password Scheme The USENIX Association*, 1999, 13 s. Dostupné z URL: <https://www.usenix.org/legacy/event/usenix99/provos/provos.pdf>.

[22] RIVEST, R. L., SHAMIR, A., and ADLEMAN, L. *A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM*, s. 120-126, 1978.

[23] DIERKS, T., RESCORLA, E. *RFC 5246. The Transport Layer Security (TLS) Protocol Version 1.2.* The Internet Engineering Task Force, 2008.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

AES	štandard pokročilého šifrovania – Advanced Encryption Standard
CA	certifikačná autorita
CBC	Cipher Block Chaining
CRL	zoznam zrušených certifikátov – certificate revocation list
DN	jedinečné meno – distinguished name
DSA	algoritmus digitálneho podpisu – digital signature algorithm
ECC	kryptografia eliptických kriviek – elliptic curves cryptography
eIDAS	nariadenie Európskej únie – electronic IDentification, Authentication and trust Services
EJBCA	Enterprise Java Beans Certificate Authority
EÚ	Európska únia
GNU GPL	všeobecná verejná licencia GNU – GNU General Public License
HSM	hardvérový bezpečnostný modul – hardware security module
HSQLDB	Hyper SQL database
JSP	JavaServer Pages
NIST	Národný inštitút pre štandardy – National Institute of Standards and Technology
NSA	Národná Bezpečnostná Agentúra – National Security Agency
OCSP	protokol na on-line overenie platnosti certifikatu – online certificate status protocol
OID	identifikátor objektu – object identifiers
openJDK	Open Java Development Kit
PKCS	public-key cryptography standards
PKI	infraštruktúra verejných kľúčov – public key infrastructure
RA	registračná autorita
RSA	algoritmus autorov Rivest, Shamir a Adleman
SHA	secure hash algorithm
SCEP	Simple Certificate Enrollment Protocol
SSL	Secure sockets layer
TLS	Transport layer security
TSA	autorita časových pečiatok – timestamping authority
TSP	Time stamping protocol

A OBSAH PRILOŽENÉHO CD

Na priloženom CD sa nachádza PDF verzia diplomovej práce s názvom DenisHerinekDP.pdf.